AÑO III - NUM. 66

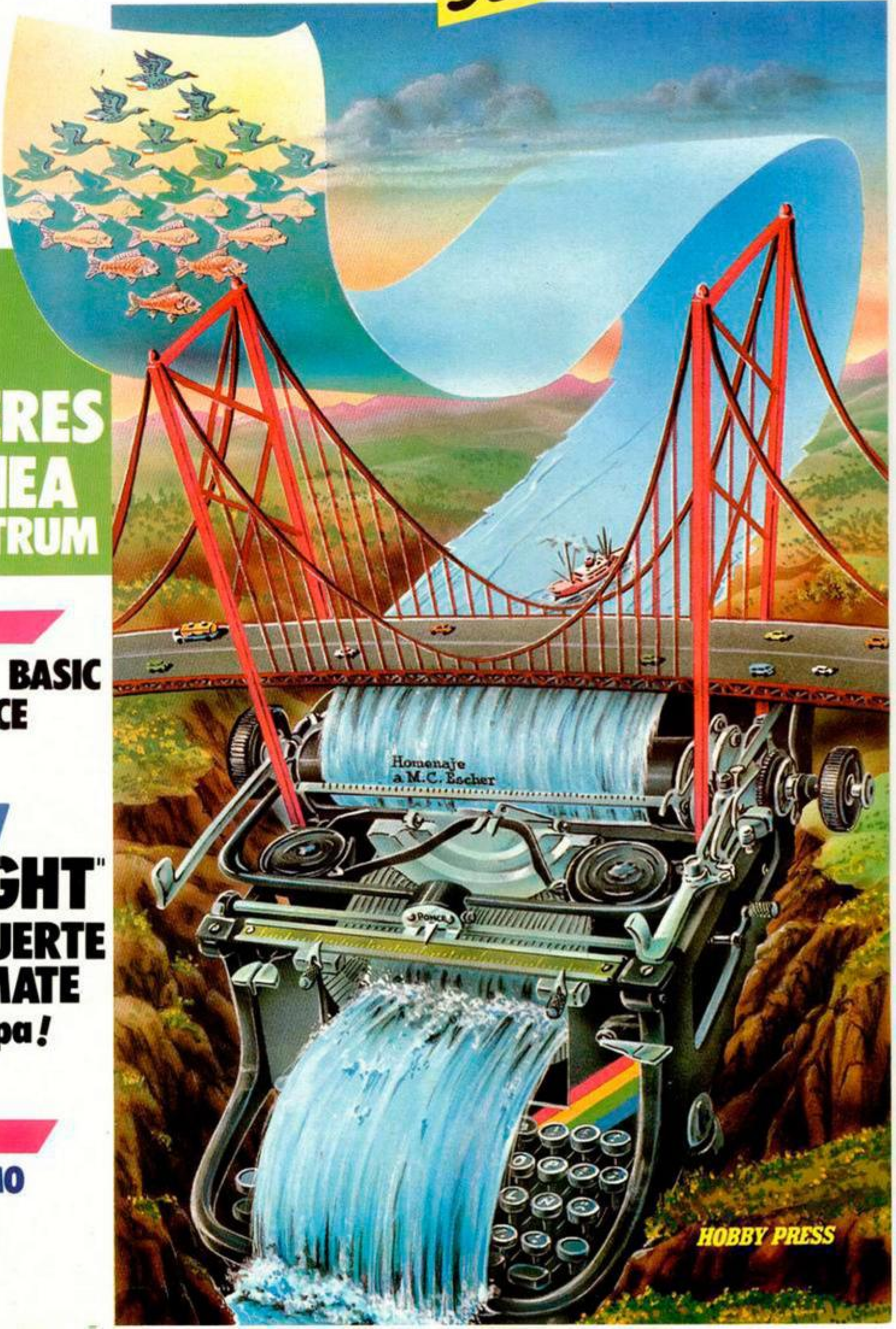
MICROHOBBY

SEMANAL

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR

AÑO III - N.º 66

135 PTS.
Canarias 140 ptas.



PROFESOR PARTICULAR
SISTEMAS DE ECUACIONES NO LINEALES

UTILIDADES
42 CARACTERES POR LÍNEA EN TU SPECTRUM

MICROFILE
MANEJO DESDE BASIC DEL INTERFACE BETA DISK

NUEVO
"GUNFRIGHT" DUELO A MUERTE CON ULTIMATE ...incluye mapa!

EXPANSION
SACA EL MÁXIMO PARTIDO A TU IMPRESORA

HOBBY PRESS

SI BUSCAS LO MEJOR **ERBE** *Software* LO TIENE



BATTLE OF PLANETS ES EL PROGRAMA SELECCIONADO PARA EL
CAMPEONATO INTERNACIONAL DE JUEGOS DE ORDENADOR.
CONSIGUE LA MAXIMA PUNTUACION ESPAÑOLA Y PARTICIPA EN LA FINAL DE LONDRES.
(LAS BASES DEL CONCURSO ESTAN EXPLICADAS EN EL PROGRAMA)

DISTIBUIDOR EXCLUSIVO PARA ESPAÑA ERBE SOFTWARE C/. STA. ENGRACIA, 17. 28010 MADRID, TFNO.: (91) 447 34 10
DELEGACION BARCELONA, AVDA. MISTRAL, N.º 10 - TFNO.: (93) 432 07 31

MICROHOBBY ESTA SEMANA

AÑO III. N.º 66. 18 al 24 de febrero de 1986
135 ptas.





«Gunfright», el último programa de Ultimate que hará vuestras delicias. Pág. 12.

Director Editorial
José I. Gómez-Centurión

Director Ejecutivo
Domingo Gómez

Asesor Editorial
Gabriel Nieto

Redactor Jefe
Africa Pérez Tolosa

Diseño
Rosa María Capitel

Redacción
Amalio Gómez, Pedro Pérez, Jesús Alonso

Secretaria Redacción
Carmen Santamaría

Colaboradores
Primitivo de Francisco, Rafael Prades, Miguel Sepúlveda Sergio Martínez y J. M. Lazo

Corresponsal en Londres
Alan Heap

Fotografía
Javier Martínez, Carlos Candel

Portada
José María Ponce

Dibujos
J. R. Ballesteros, A. Perera, F. L. Frontán, Pejo, J. M. López Moreno, J. Igual, J. A. Calvo, Lóriga, J. Olivares

Edita
HOBBY PRESS, S. A.

Presidente
María Andrino

Consejero Delegado
José I. Gómez-Centurión

Jefe de Publicidad
Marisa Esteban

Publicidad Barcelona
José Galán Cortés
Tels.: 303 10 22 - 313 71 76

Secretaria de Dirección
Marisa Cogorro

Suscripciones
M.ª Rosa González
M.ª del Mar Calzada

Redacción, Administración y Publicidad
La Granja, 39
Polígono Industrial de Alcobendas
Tel.: 654 32 11
Telex: 49480 HOPR

Dto. Circulación
Carlos Peropadre

Distribución
Coedis, S. A. Valencia, 245
Barcelona

Imprime
Rotedic, S. A. Ctra. de Irún, km. 12,450 (MADRID)

Fotocomposición
Espacio y Punto, S. A.
Paseo de la Castellana, 268

Fotomecánica
Grof
Ezequiel Solana, 16

Depósito Legal
M-36.598-1984

Representante para Argentina, Chile, Uruguay y Paraguay, Cía. Americana de Ediciones, S.R.L. Sud América 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina).

Solicitado control
OJD

PREMIADOS HOBBY-SUERTE

CASIMURA ESCUDERO AUTOCAR. Cuevas, 8 (MADRID).
Cinta de programas (5.ª Cat.)

JOSE M.ª BORAN CEBRIAN. Primo de Rivera, 5. Jaca (HUESCA).
Cinta de programas (5.ª Cat.)

ATAULFO MORENO LOPEZ. Rosalía de Castro, 3, 1.º. Alcobendas (MADRID).
Cinta de programas (5.ª Cat.)

JOSE LUIS SIMON LLANO. Ensireza, 2, 6.º (MADRID).
Suscripción a Microhobby Semanal por un año (4.ª Cat.)

MIGUEL A. ARIÑA. Pintor Picasso, 3. Móstoles (MADRID).
Joystick con su Interface (3.ª Cat.)

CONCEPCION ESPINOSA IALADO. Pto. de las Mazalomas, 24, 3 (MADRID).
Cinta de programas (5.ª Cat.)

JOSE LUIS ALONSO BOYER. Juan de Archeta, 16, 7.º (BILBAO).
Cinta de programas (5.ª Cat.)

BENJAMIN BARCENILLA. Los Monegros, 26. Zarzaquemada (MADRID).
Suscripción a Microhobby Semanal por un año (4.ª Cat.)

CONCEPCION LIMA SANTOS. Plaza Fuensagrada, 5 (MADRID).
Joystick con su Interface (3.ª Cat.)

MARIANO MORENO QUIJADA. S. Joaquín, 32. Cieza (MURCIA).
Un Spectrum 48k (1.ª Cat.)

ALEJANDRO S. MARTIN ARTECHE. Pelayo, 52 (MADRID).
Cinta de programas (5.ª Cat.)

PEDRO SANCHEZ PALLERO. Artesanos, 14, 3.º (HUELVA).
Cinta de programas (5.ª Cat.)

JOSE L. DOMINGUEZ VILLAREJO. Fray José de Terbeiña, 15 (MADRID).
Cinta de programas (5.ª Cat.)

MANUEL ROMERO FRAIRIA. Mero, 1, 5.º (SEVILLA).
Cinta de programas (5.ª Cat.)

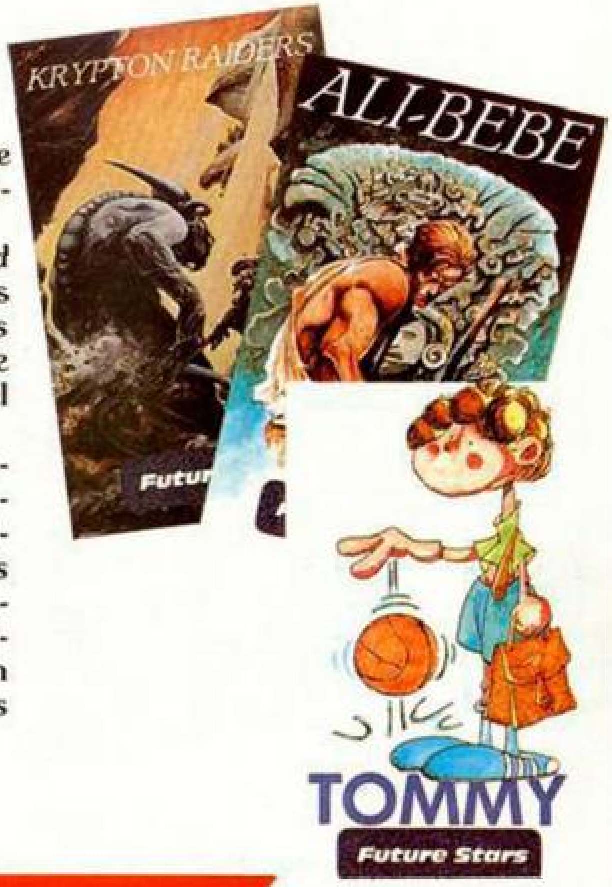
JUAN JOSE CABALLERO MUÑOZ. Antonio Ulloa, 4, 4.º (CADIZ).
Cinta de programas (5.ª Cat.)

ANTONIO RIBAS MARIA. S. Roque, 18, 1.º (GUADALAJARA).
Cinta de programas (5.ª Cat.)

IÑAKI FERNANDEZ DE RETAMA. Sancho el Sabio, 15, Int. 3.º (VITORIA).
Cinta de programas (5.ª Cat.)

MICROPANORAMA

LA FUTURA ESTRELLA DE DINAMIC



El próximo éxito de Dinamic ya tiene nombre: Phantomas. Este nuevo programa versará sobre las aventuras y desventuras de un poco agraciado ladrón interespacial que se las verá y deseará para conseguir llegar hasta una caja fuerte, hacerse con lo que se encuentra en su interior y escapar después del acoso de los policías galácticos que le perseguirán incansablemente.



Pero esta no es la única novedad que Dinamic tiene preparada para las próximas fechas, ya que otros tres títulos pertenecientes al sello Future Stars harán pronta aparición en el mercado.

Future Stars reúne una serie de juegos realizados por programadores noveles y que tiene como objetivo principal el estimular la creación de nuevos programas. Estos primeros juegos serán: Tommy, Krypton, Raiders y Ali-Bebé, los cuales dejan entrever el gran talento programativo de estas futuras estrellas de la programación.

DESERT RATS

Más diversión para los estrategas

En las próximas fechas saldrá a la venta un nuevo título de la colección, que bajo el nombre de Juegos & Estrategia, reúne una serie de títulos relacionados con los juegos de simulación estratégica y de inteligencia.

En esta ocasión se trata de Desert Rats, (Ratas del desierto), un magnífico war-game realizado por Cases Computers Simulations y que nos hará protagonistas de los acontecimientos que tuvieron lugar en la campaña «Oeste del Desierto» en la cual los Afrika Corps fueron derrotados por las tropas de Eje en la famosa batalla de Alamein.





Este war-game está muy en la línea de anteriores programas como Arnhem, pero incorpora una serie de novedades, tanto en el aspecto estratégico como en el desarrollo en sí del juego, que hacen de él un programa verdaderamente completo e interesante y que hará las delicias de todos los aficionados a los juegos de estrategia.

EL BASIC DEL SPECTRUM CORRERA EN EL ENTERPRISE

Una prueba del lugar tan relevante que ocupa el Spectrum en el mundo de los ordenadores, así como de la excelente calidad de su software es la realización por parte de Enterprise, una conocida casa de ordenadores, del programa Basic to Basic.

Este programa permite, una vez que está corriendo en el Enterprise, cargar cualquier programa realizado en el Basic del Spectrum, convirtiéndolo en el Basic propio de este ordenador.

Su precio en el mercado británico es de 20 libras, y ofrece a los usuarios de Enterprise la posibilidad de disfrutar las grandes ventajas de este maravilloso ordenador.

INVESTRONICA LANZA SUS NUEVOS MODELOS HOMOLOGADOS



Investrónica, distribuidora oficial en España de Sinclair, está lanzando al mercado los nuevos modelos de ordenadores con las rectificaciones oportunas para cumplir las recientes normas de homologación que, en fechas recientes, el Ministerio de Industria exigió para la venta y distribución de estos productos en nuestro país.

Este es el caso del Spectrum +, que ya está disponible en los comercios con el teclado en castellano. Este modelo en su versión anterior ha sido el más vendido por Investrónica en la pasada campaña Navideña, aunque aún Investrónica no ha facilitado las cifras oficiales de las ventas alcanzadas por todos sus productos.

En números siguientes ofreceremos estas cifras así como un estudio más detallado de el nivel de ventas y del parque estimativo de Sinclair en nuestro país.

ULTIMATE DEJARA DE DISTRIBUIR SUS PROGRAMAS



Ultimate, una de las casas de software más prestigiosas en todo el mundo, por no decir la que más, ha decidido en las últimas fechas dedicarse exclusivamente a la creación de programas y dejar a un lado todos los temas referentes a la distribución y venta de sus productos.

A partir de ahora, otra no menos famosa compañía, U.S. Gold, será su representante en exclusiva en todo el mundo.

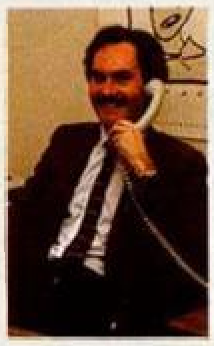
Pero debido a que la distribuidora en España U.S. Gold es ERBE, se deduce que los programas de Ultimate serán distribuidos en nuestro país por esta compañía.

Como primeros lanzamientos ERBE cuenta con los siguientes programas: Alien 8 y Knight Lore para MSX y, Cyberum para Spectrum.

Los dos primeros programas ya son suficientemente conocidos por todos nuestros lectores, pero Cyberum representa una auténtica novedad.

Este juego consiste en una carrera espacial para poder controlar el Universo, el Cyberum. Esta dura competición tiene como objetivo fundamental llegar hasta Zebarema, un pequeño sistema planetario que es el principal productor de Cybernita, la sustancia más resistente del Universo y que permite a los que la poseen controlar la más importante fuente de energía: las estrellas.

AQUI LONDRES



En el Which Computer Show de Birgimham, Commodore acaparó toda la atención al lanzar su nuevo 128 D, y al exhibir su Amiga por primera vez. El 128 D está destinado a ser un complemento del 128 para pequeños negocios; entre sus características principales se incluyen un disc-drive de 4,5 pulgadas incorporado y un 1.571 double disc-drive, a la vez que un teclado independiente y portátil que atraerá a los usuarios profesionales. El conjunto de 128 D incluye también, un monitor de fósforo verde y tiene dos modalidades de 40 y 80 columnas. Se ha proyectado que este ordenador salga al mercado el mes próximo, al precio de 540 £.

El ordenador Amiga atrajo a multitud de público, que se encontró con una sorpresa: le acompañaba poco software interesante para los profesionales, que no hay que olvidar, representan el mercado al que este ordenador va dirigido.

En una encuesta recientemente realizada en Inglaterra el juego más popular de 1985, fue considerado «Elite», de Firebird, seguido a muy escasa distancia por «Way of the Exploding Fist», de Melbourne House: El premio para el ordenador más interesante fue para el Amstrad 6128, compañía que ganó este premio el año pasado con el CPC 464.

Atari y Commodore han determinado atacar el lucrativo mercado educativo. Atari ha ofrecido a colegios e institutos un descuento de 150 £ con la compra del 520 ST, el cual vino acompañado de un disc-drive, monitor y ratón, más una selección de software.

Commodore espera que el 128 D, con un precio similar al Atari 520 ST, resulte atractivo para los colegios aún sin incentivos. Sin embargo, tienen dudas en cuanto llegar al anillo comercial que Acorn (a través de la gama BBC) y Reserch Machine, poseen en la actualidad. Aunque sus productos han sido catalogados como de «vieja tecnología» por sus competidores, instituciones educativas se mostrarán reticentes a llevar a cabo un cambio drástico en sus hábitos de compras de Hardware. De hecho, existe una satisfacción general con los productos de Acorn y Resech Machine y un cambio se plantea casi imposible. Sin embargo, no hay duda del gran avance que el 520 ST y el Amiga tienen sobre sus predecesores en lo que respecta a la capacidad de memoria y poder de posibilidades programativas.

Un dato importante: un gran número de compañías de software tanto en U.S.A. como en U.K. en la actualidad están desarrollando programas para estos ordenadores de 16 bites, ambos basados en el procesador 68.000 y aunque estos programas son relativamente caros de momento (alrededor de 20 £), se esperan reducciones de precio.

De nuestro corresponsal en Londres ALAN HEAP

TRUCOS

ALINEAR NUMEROS

Juan Grimbérieux nos manda un truco nada menos que desde Bélgica y, concretamente desde Lieja, que nos sirve para alinear números y, según él mismo nos indica, para mejorar el programa que nos ofrecía José Ramón Hernández Luque en un truco publicado en el número 35 de Microhobby. Con esta mejora se lleva a cabo la alineación, aún con potencias exactas de 1Ø.

```
  10 INPUT "PUNTO DE TABULACION
DE UNIDADES ?",A
  20 INPUT "DIGITAR NUMERO",N
  30 LET A$=STR$ N
  40 LET E=LEN A$
  50 PRINT TAB A-E+1;N: GO TO 20
```

EN ESPAÑOL

Cuando queremos grabar un programa en cinta, nos sale la sentencia SAVE "nombre" del programa en cuestión, como todos sabéis, y, a continuación, Star tape, Press any key. Pues bien, para que esta última sentencia sea legible para todos, es decir, para que salga en español, sólo tendremos que teclear el programa que nos ha mandado un lector anónimo, este que os ofrecemos a continuación.

```
  20 BORDER 0: PAPER 0: INK 7 .
  30 POKE 23624,64: CLS
  40 PRINT #1; INK 7;"Casette en
grabacion,pulsa Tecla"
  50 SAVE "nombre"
  60 BORDER 0
```

AUMENTAR EL NUMERO DE Ks

Muchas veces nos han preguntado si es posible ejecutar un programa para 48 K en un Spectrum de 16. Pues bien, respecto a este tema Manuel Zumaquero nos manda unos trucos con los que esto será posible.

El primero, muy sencillo, consiste en grabar en dos partes el programa, por ejemplo, en una las instrucciones y los gráficos, y en otra el programa en sí. Para hacerlo más estético se puede poner al principio del primer programa: PRINT " PARE LA CINTA ", y al final de éste: PRINT " PULSE PLAY ":LOAD "".

Otro método, un poco más sofisticado, es mediante una rutina en Código Máquina:

```
  10 LET c=0: FOR n=23300 TO 234
21
  20 READ x: LET c=c+x: POKE n,x
  30 NEXT n
  35 IF c<>11606 THEN PRINT "err
or"
  40 DATA 42,83,92,43,237,75,75,
92,35,167,237,66,9,200,35
  50 DATA 35,78,35,70,229,35,126
,254,13,32,3,209,24,231,254
  60 DATA 14,32,243,209,213,11,1
1,11,197,120,18,27,121,18,229
  70 DATA 43,126,254,47,40,19,25
4,46,56,15,254,196,40,4,254
  80 DATA 56,48,7,35,35,119,43,4
3,24,231,35,54,176,35,54
  90 DATA 34,225,35,35,54,34,35,
84,93,213,35,35,35,229,42
 100 DATA 89,92,167,237,82,68,77
,225,237,176,42,75,92,43,43
 110 DATA 43,34,75,92,42,89,92,4
3,43,43,34,89,92,225,193,24,155,
0,0,0
```



Cuando se termine de copiar debemos ejecutarlo y responder varias veces al mensaje de Scroll? hasta que aparezca el mensaje Ø,Ø O.K. Debemos salvar el programa con la instrucción SAVE "algún nombre" CODE 233ØØ,122. Después de esto, podremos utilizarlo, siempre que borremos las líneas de este programa, que podrían producir un crack.

Cuando tengamos el programa en Basic y queramos comprimirlo deberemos cargar este programa y teclear RANDOMIZE USR 233ØØ (RAND USR 233ØØ). Veremos que después de aparecer el mensaje Ø,Ø O.K., el listado se ha modificado introduciendo en cada número, VAL "número". De esta forma ahorraremos el 14% de la memoria utilizada.

En este espacio también tienen cabida los trucos que nuestros lectores quieran proponer.
Para ello, no tienen más que enviarlos por correo a MICROHOBBY, C/ La Granja, 8. Polígono Industrial de Alcobendas (Madrid).

LA MAQUINA DEL TIEMPO

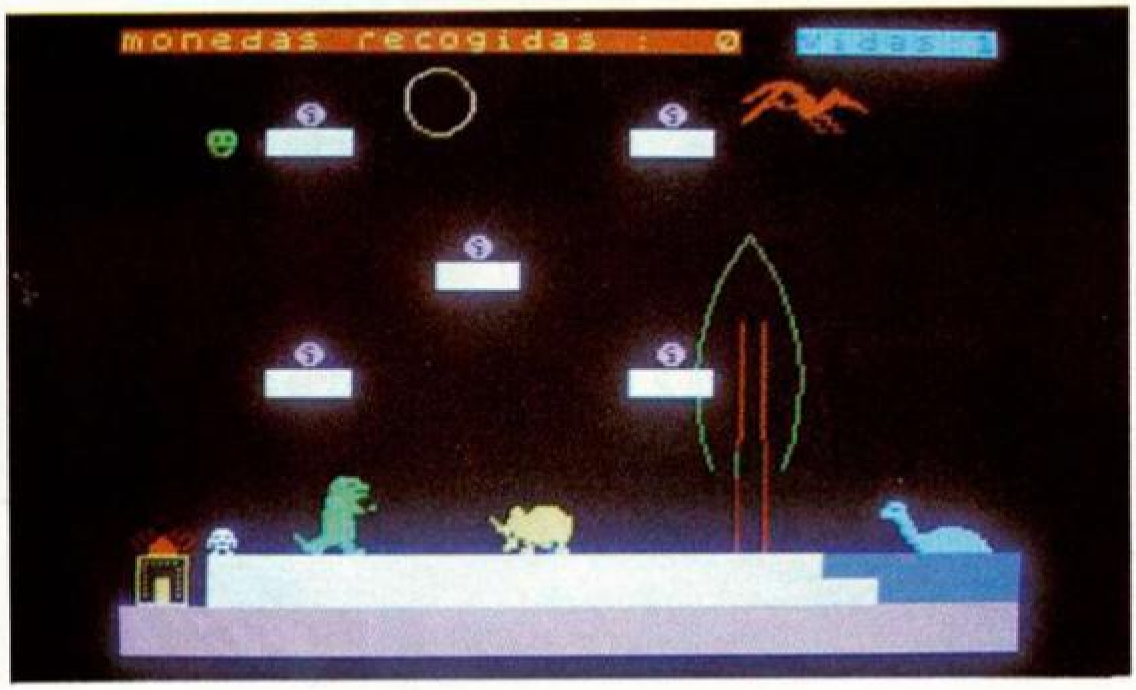
Rafael MARQUEZ

Spectrum 48 K

Un afamado ladrón de bancos acaba de cometer su más extravagante y alocada fechoría: robar el mundialmente conocido Banco Central y escapar con el científico «Loquillo» a través del Tunel del Tiempo.

Este invento, el último cuajado en la retorcida mente del científico, les teletransportará por diferentes épocas y conseguir atraparles puede ser algo más que imposible.

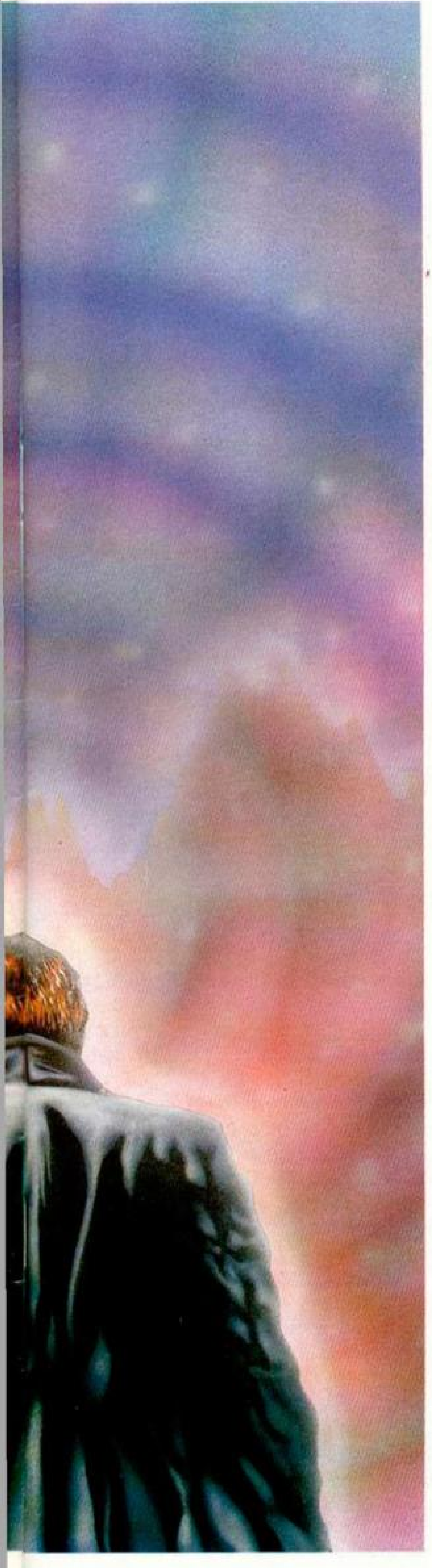
Ante esta situación, tú, experto rastreador, tendrás que aventurarte en la arriesgada, y también tendadora, misión de dar con sus huesos enfrentándote, claro está, con lo desconocido. ¿Será demasiado para ti? ¿Lo conseguirás...?



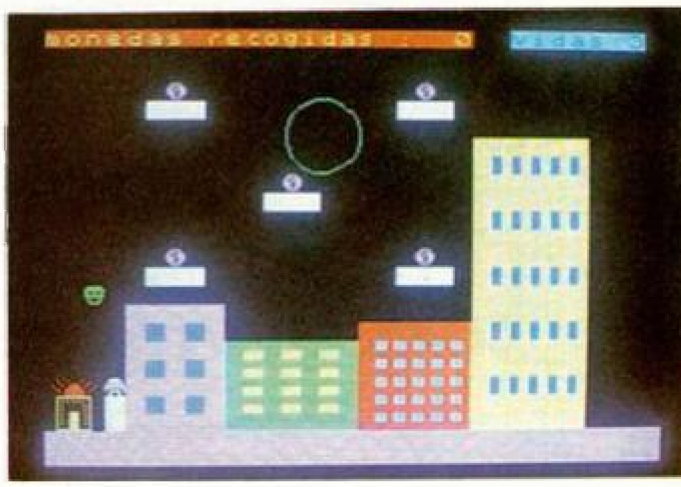


```
   3 GO SUB 8000
   5 LET pant=1: LET m=0: LET v=
2
   8 LET vid=z-48
  20 LET a$="C"
 100 GO SUB 7000+pant*100-100
 110 GO SUB 9900: GO SUB 7700
 120 LET x=19: LET y=3
 140 PRINT AT 0,0; INK 6; PAPER
2;"monedas recogidas : ";m;AT 0
,24; PAPER 5; INK 1;"vidas ";vid
 150 LET a=INT (RND*20): LET b=I
NT (RND*20)
 160 PRINT AT 20,3; BRIGHT 1;"█"
;AT 21,3;"█"
 499[redacted]
 500 LET a=a+SGN (x-a)/2: LET b=
b+SGN (y-b)/v
 505 IF ATTR (x+1,y)=71 THEN LET
 x=x-1
 510 IF x<21 THEN LET x=x+1
 515 IF INKEY$<>"1" THEN IF INKE
Y$<>"2" THEN LET a$="C"
 520 IF INKEY$="0" THEN IF ATTR
(x-2,y)<>71 THEN IF x>2 THEN LET
 x=x-2
 530 IF INKEY$="1" THEN LET a$="
E": IF ATTR (x,y-1)<>71 THEN IF
y>0 THEN LET y=y-1
 540 IF INKEY$="2" THEN LET a$="
D": IF ATTR (x,y+1)<>71 THEN IF
y<31 THEN LET y=y+1
 545 IF x=21 AND y=2 THEN GO TO
4000
 550 IF ATTR (x,y)=3 THEN LET m=
m+1: PRINT AT x,y;" ";AT 0,21; I
NK 6; PAPER 2;m: FOR g=20 TO 25:
 BEEP .01,g: NEXT g: IF m/5=INT
(m/5) THEN GO SUB 3000
 560 IF ATTR (x,y)<>7 THEN GO TO
 2000
1000 PRINT AT x,y;a$
1005 IF ATTR (a,b)=7 THEN PRINT
AT a,b; INK 4;"B"
1010 PAUSE 2: BEEP .01,10
1020 IF x=a AND y=b THEN GO TO 2
000
1030 PRINT AT x,y;" "
1040 IF ATTR (a,b)=4 THEN PRINT
AT a,b;" "
1900 GO TO 500
1999[redacted]
2000 LET vid=vid-1: PRINT AT 0,3
0; INK 1; PAPER 5;vid: BEEP .1,-
20
2010 FOR g=0 TO 10: FOR e=0 TO 7
: PRINT AT x,y; INK e;"C": BEEP
.01,g+10: BEEP .01,e+20: NEXT e:
 NEXT g
2020 IF vid=0 THEN GO TO 2500
2030 LET m=(pant-1)*5
2040 CLS : GO TO 20
2499[redacted]
2500 BORDER 1: PAPER 6: INK 1: C
LS
2510 PRINT AT 5,10; FLASH 1;"!LO
 SIENTO!";AT 10,9;"HAS FRACASADO
"
2520 FOR g=10 TO 0 STEP -1: FOR
e=0 TO -20 STEP -1: BEEP .01,g:
BEEP .01,e: NEXT e: NEXT g
2530 PRINT AT 15,7;"OTRA PARTIDA
 (S/N)?"
2540 IF INKEY$="s" THEN RUN
2550 IF INKEY$="n" THEN STOP
2560 GO TO 2540
2999[redacted]
3000 PRINT AT 20,3;" ";AT 21,3;"
"
3100 PRINT BRIGHT 1;AT 19,0; FLA
SH 1; INK 2; PAPER 0;"FGH";AT 20
,0; FLASH 0; INK 6;"IJK";AT 21,0
;"L"; INK 5; FLASH 1;"B"; FLASH
0; INK 6;"N"
3200 RETURN
3999[redacted]
4000 FOR g=0 TO 30: BORDER RND*7
: BEEP .01,RND*50: NEXT g
4010 FOR g=0 TO 21: PRINT AT g,0
; OVER 1; INK RND*7;"[redacted]": BEEP .01
,g: NEXT g
4020 FOR g=0 TO 30: BORDER RND*7
```

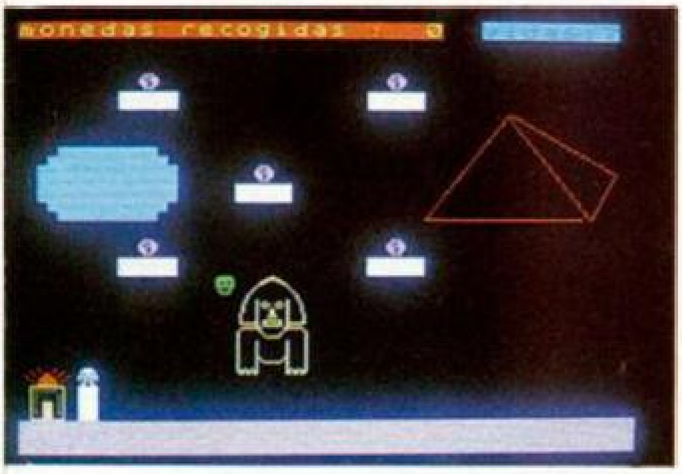
JAVIER IGUAL



```
: BEEP .01,RND*50: NEXT g
4025 FOR g=0 TO 300: NEXT g
4026 BORDER 0: FOR g=0 TO 21: PRINT OVER 1; INK 6; PAPER 1;AT g,0;"████████████████████████████████": NEXT g: FOR g=0 TO 300: NEXT g
4030 LET pant=pant+1: IF pant=7 THEN GO TO 6000
4040 GO TO 100
5999 REM [illegible]
6000 BORDER 6: PAPER 2: INK 7: CLS
6010 PRINT AT 5,4;"Has atrapado al ladron y hasrecuperado el botin."
6020 PRINT AT 10,10; FLASH 1;"!ENHORABUENA!"
6030 FOR g=0 TO 1: FOR e=0 TO 30: BEEP .01,e: BEEP .01,e-10: BEEP .01,e+10: NEXT e: NEXT g
6040 FOR g=0 TO 10: FOR e=20 TO 25: BEEP .01,e: NEXT e: NEXT g
6050 PRINT AT 15,7; PAPER 7; INK 1;"OTRA PARTIDA (S/N)?"
6060 IF INKEY$="s" THEN RUN
6070 IF INKEY$="n" THEN STOP
6080 GO TO 6060
6999 REM primera epoca
7000 BORDER 3: PAPER 0: INK 7: CLS : PRINT AT 10,10; FLASH 1;"EDAD DE PIEDRA": GO SUB 7800: CLS : GO SUB 9000
7001 PRINT AT 17,7; BRIGHT 1; INK 4;"AB";AT 18,7;"CD";AT 19,6;"EFG"
7002 PRINT INK 6;AT 18,13;"HIJ";AT 19,13;"KLM"
7004 PRINT INK 5; BRIGHT 1;AT 18,27;"NO";AT 19,27;"PQRS"
7006 PRINT AT 20,3; BRIGHT 1;"██████████████"; INK 1;"███"; INK 7;AT 21,3;"█████████"; INK 1;"████"
7007 GO SUB 9025
7009 PRINT INK 2; BRIGHT 1;AT 2,22;"ABCDE";AT 3,22;"FGHIJ"
7010 CIRCLE BRIGHT 1; INK 6;88,152,10
7012 BRIGHT 1: INK 2: PLOT 175,16: DRAW 1,70: PLOT 183,16: DRAW -1,70
7014 INK 4: PLOT 178,112: DRAW -10,-70,1: PLOT 178,112: DRAW 10,-70,-1: BRIGHT 0: INK 7
7090 RETURN
7099 REM segunda epoca
7100 BORDER 3: PAPER 0: INK 7: CLS : PRINT AT 10,14; FLASH 1;"EGIPTO": GO SUB 7800: CLS
7110 GO SUB 9040
7120 PRINT INK 6; BRIGHT 1;;AT 14,11;"ABCD";AT 15,11;"EFGH";AT 16,11;"IJKL";AT 17,11;"MNOP";AT 18,11;"MQRP";AT 19,11;"STO"
7130 GO SUB 9080: PRINT AT 19,14; INK 6; BRIGHT 1;"A"
7132 PRINT INK 5;AT 7,1;"████";AT 8,1;"█████";AT 9,1;"██████";AT 10,1;"████"
7136 BRIGHT 1: INK 2: PLOT 232,88: DRAW -64,0: DRAW 34,45: DRAW 34,-45: DRAW 10,20: DRAW -43,26: BRIGHT 0: INK 0
7190 RETURN
7199 REM tercera epoca
7200 BORDER 3: PAPER 0: INK 7: CLS : PRINT AT 10,10; FLASH 1;"IMPERIO ROMANO": GO SUB 7800: CLS
7210 GO SUB 9100
7220 PRINT INK 6; BRIGHT 1;AT 8,15;"ABCDE";AT 9,15;"FGHIJ";AT 10,15;"KLMNO"
7230 PRINT INK 4; BRIGHT 1;AT 10,8;"PQ";AT 11,8;"RS"
7240 INK 2: BRIGHT 1: PLOT 80,120: DRAW 0,32: DRAW 24,0: DRAW 0,-32: DRAW -8,0: DRAW 0,16: DRAW -8,0,2: DRAW 0,-16: DRAW -8,0
7242 PRINT AT 7,4; INK 5;"TU"
7250 INK 3: PLOT 39,80: DRAW BRIGHT 1;0,31: PLOT 40,80: DRAW BRIGHT 1;0,31
7260 INK 7: BRIGHT 0
7290 RETURN
7299 REM cuarta epoca
7300 BORDER 3: PAPER 0: INK 7: CLS : PRINT AT 10,12; FLASH 1;"EDAD MEDIA": GO SUB 7800: CLS
7310 LET v=1
7320 INK 1: BRIGHT 1: PRINT AT 5,1;"████";AT 6,1;"████": FOR g=7 TO 13: PRINT AT g,1;"████": NEXT g: PRINT AT 7,2; PAPER 1; INK 7;"█";AT 9,2;"█";AT 12,2;"██";AT 13,2;"██"
7330 INK 4: PRINT AT 5,21;"████";AT 6,21;"████": FOR g=7 TO 13: PRINT AT g,21;"████": NEXT g: PRINT AT 7,23; INK 5; PAPER 4;"█";AT 9,23;"█";AT 12,22;"██";AT 13,22;"██"
7340 INK 6: PRINT AT 11,9;"███";AT 12,9;"██████";AT 13,9;"██████": FOR g=14 TO 16: PRINT AT g,9;"████████": NEXT g
7350 PRINT AT 12,10; INK 2; PAPER 6;"█";AT 12,15;"█";AT 14,10;"█";AT 14,15;"█";AT 15,12;"██";AT 16,12;"██"
7360 INK 7: BRIGHT 0
7390 RETURN
7399 REM [illegible]
7400 BORDER 3: PAPER 0: INK 7: CLS : PRINT AT 10,11; FLASH 1;"EDAD MODERNA": GO SUB 7800: CLS
7405 BRIGHT 1
7410 FOR g=15 TO 21: PRINT INK 3;AT g,4;"█████": NEXT g: FOR g=16 TO 20 STEP 2: PRINT AT g,5; INK 1; PAPER 3;"█ █": NEXT g
7420 FOR g=17 TO 21: PRINT AT g,9; INK 4;"███████": NEXT g: FOR g=17 TO 20: PRINT PAPER 4; INK 6;AT g,10;"█ █ █": NEXT g
7430 FOR g=16 TO 21: PRINT INK 2;AT g,16;"██████": NEXT g: FOR g=17 TO 21: PRINT INK 5; PAPER 2;AT g,17;"█████": NEXT g
7440 FOR g=6 TO 21: PRINT INK 6;AT g,22;"██████": NEXT g: FOR g=7 TO 19 STEP 3: PRINT AT g,23; INK 1; PAPER 6;"█████": NEXT g
7450 CIRCLE INK 4; BRIGHT 1;112,128,16
7460 BRIGHT 0
7490 RETURN
7499 REM [illegible]
7500 BORDER 3: PAPER 0: INK 7: CLS : PRINT AT 10,12; FLASH 1;"SIGLO XXV": GO SUB 7800: CLS
7510 GO SUB 9150
7520 BRIGHT 1: FOR g=0 TO 100: PLOT INT (RND*255),INT (RND*175): NEXT g
7530 LET n$="ABC": PRINT AT 9,15; INK 6;n$;AT 15,3; INK 4;n$
7535 LET n$="DEF": PRINT INK 5;AT 11,8;n$;AT 21,7; INK 3;n$
7540 CIRCLE INK 6; BRIGHT 1;200,104,32
7545 CIRCLE INK 1; BRIGHT 1;112,48,16: CIRCLE INK 3; BRIGHT 1;80,136,8
```




```
7550 CIRCLE INK 2; BRIGHT 1;32,104,20: CIRCLE INK 2; BRIGHT 1;128,128,4
7560 BRIGHT 0
7590 RETURN
7699 REM maquina del tiempo
7700 PRINT AT 19,0; INK 2;"EGH";AT 20,0; INK 6;"IJK";AT 21,0;"LMN"
7710 INK 3: PRINT AT 3,6;"A";AT 3,19;"A";AT 8,12;"A";AT 12,6;"A";AT 12,19;"A": INK 7
7720 LET s$="███"
7730 PRINT AT 4,5; BRIGHT 1;s$;AT 4,18;s$;AT 9,11;s$;AT 13,5;s$;AT 13,18;s$
7750 RETURN
7799 REM cuenta atras
7800 FOR g=9 TO 0 STEP -1: PRINT AT 12,16;g: PAUSE 10: BEEP .01,g: PAUSE 10: BEEP .01,g-10: PAUSE 10: BEEP .01,g+10: NEXT g: FOR g=0 TO 30: BEEP .005,g: NEXT g: RETURN
7999 REM instrucciones
8000 BORDER 4: PAPER 0: INK 1: BRIGHT 1: CLS
8010 PRINT AT 0,0; INK 2; PAPER 6;"          INSTRUCCIONES          "
8015 PRINT AT 2,0; PAPER 7; INK 4;"      *La Maquina del Tiempo*      "
8020 PRINT AT 4,0; PAPER 5;"  Tienes que atrapar al la- dron del Banco Central que,juntocon un cientifico loco ha esca- pado en La Maquina del Tiempo.  Para conseguirlo debes vagar pordistintas epocas recogiendo las monedas que encuentres y cuando ya no queden mas,entrar en la   maquina que te teletransportara a otra epoca.  "
8025 PRINT : PRINT PAPER 5;"  No toques nada de lo que  veas en tu viaje por que podria ser peligroso.  "
8027 PRINT : PRINT PAPER 5;"Controles: 1-Izq. 2-Derc. 0-Sub."
8030 PRINT : PRINT INK 3; PAPER 7;"  Cuantas vidas quieres (1-9)?  "
8035 BRIGHT 0: FLASH 0
8040 LET z=CODE INKEY$
8050 IF z<49 OR z>57 THEN GO TO 8040
8060 FOR g=0 TO 20: BEEP .02,g: BEEP .02,g-20: NEXT g
8070 CLS
8100 RETURN
8999 REM [illegible]
9000 RESTORE 9010: FOR g=0 TO 151: READ a: POKE USR "a"+g,a: NEXT g: RETURN
9010 DATA 15,31,62,63,63,63,127,127,192,224,120,248,120,248,192,128,127,127,255,255,255,255,255,255,194,230,240,0,0,128,128,128,0,128,193,103,63,63,31,7,255,255,255,255,247,135,3,3,128,128,128,0,128,128,240,240
```




```
9015 DATA 0,1,3,3,131,199,103,63
,0,135,207,223,255,255,255,255,0
,240,248,252,255,255,255,255,63,
13,5,7,7,15,12,0,255,255,255,63,
55,25,120,120,255,255,254,254,20
4,204,30,30
9020 DATA 60,62,230,254,255,15,7
,3,0,0,0,0,0,0,128,128,3,3,1,0,0
,0,0,0,192,195,255,255,255,63,63
,31,254,255,255,255,255,255,255,
255,0,192,240,240,252,252,254,25
5
9025 RESTORE 9030: FOR g=0 TO 79
: READ a: POKE USR "a"+g,a: NEXT
 g: RETURN
9030 DATA 0,3,7,14,60,113,227,7,
126,255,255,119,227,193,112,243,
0,128,192,192,240,249,251,255,0,
0,32,112,252,254,255,223,0,0,0,0
,0,0,128,192
9035 DATA 15,28,56,112,224,128,0
,0,63,0,0,0,0,0,0,0,255,254,253,
119,16,28,4,7,128,0,0,128,128,22
4,32,56,224,48,0,0,0,0,0,0
9040 RESTORE 9045: FOR g=0 TO 16
7: READ a: POKE USR "a"+g,a: NEX
T g: RETURN
9045 DATA 0,0,0,0,0,0,1,2,1,7,28
,48,64,128,0,0,128,224,56,12,2,1
,0,0,0,0,0,0,0,0,128,64
9050 DATA 4,4,8,8,16,16,16,16,0,
0,64,76,82,76,65,65,0,0,2,50,74,
50,130,130,32,32,16,16,8,8,8,8
9055 DATA 32,32,32,32,63,0,31,32
,65,67,64,79,200,79,224,32,130,1
94,2,242,19,242,7,4,4,4,4,4,252,
0,248,4
9060 DATA 64,64,64,64,64,64,64,6
4,16,8,7,0,64,64,64,64,8,16,224,
0,2,2,2,2,2,2,2,2,2,2,2,2,64,64,
64,96,80,79,64,64,2,2,2,6,10,242
,2,2
9065 DATA 73,118,54,0,0,0,0,0,64
,192,192,0,0,0,0,0,2,3,3,0,0,0,0
,0
9080 RESTORE 9085: FOR g=0 TO 7:
 READ a: POKE USR "a"+g,a: NEXT
g: RETURN
9085 DATA 146,110,108,0,0,0,0,0
9100 RESTORE 9105: FOR g=0 TO 16
7: READ a: POKE USR "a"+g,a: NEX
T g: RETURN
9105 DATA 0,0,0,0,3,4,7,4,1,6,56
,192,0,0,0,192,255,0,0,0,0,0,0,0
,128,96,28,3,0,0,0,3,0,0,0,0,192
,32,224,32
9110 DATA 4,4,4,4,4,4,4,4,56,6,6
5,228,238,238,14,64,0,0,255,0,68
,238,238,238,26,96,126,4,78,238,
238,224,32,32,32,32,32,32,32,32
9115 DATA 4,4,4,3,0,0,0,0,228,23
8,110,14,192,56,6,1,0,68,238,238
,238,0,0,255,4,78,238,232,195,28
,96,128,32,32,32,192,0,0,0,0
9120 DATA 67,67,67,65,255,67,67,
67,192,192,192,128,224,208,200,2
00,67,67,67,66,66,66,66,78,208,2
24,192,64,64,64,64,112
9125 DATA 1,1,3,3,3,7,7,1,128,12
8,192,192,192,224,224,128
9150 RESTORE 9155: FOR g=0 TO 47
: READ a: POKE USR "a"+g,a: NEXT
 g: RETURN
9155 DATA 0,0,0,0,1,15,127,127,0
,0,63,97,255,255,255,255,127,254
,252,240,252,252,240,240
9160 DATA 254,126,63,15,63,63,15
,15,0,0,252,135,255,255,255,255,
0,0,0,0,128,224,254,254
9900 RESTORE 9910: FOR g=0 TO 11
1: READ a: POKE USR "a"+g,a: NEX
T g: RETURN
9910 DATA 60,102,223,231,247,102
,60,0
9915 DATA 126,255,219,219,255,66
,102,60
9920 DATA 60,126,90,255,165,189,
36,102,60,126,122,126,126,60,72,
108,60,126,94,126,126,60,18,54
108,60,126,94,126,126,60,18,54
9925 DATA 0,4,2,17,8,4,2,0,129,6
6,24,60,126,255,255,255,0,32,64,
136,16,32,64,0
9930 DATA 7,4,5,4,5,4,5,4,255,0,
85,0,0,0,0,60,224,32,32,32,160,3
2,160,32
9935 DATA 5,4,5,4,5,4,4,7,60,60,
60,60,60,60,60,255,160,32,160,32
,160,32,32,224
```

CORRECCION AL PROGRAMA IGLOO

En el programa «Igloo» que publicábamos en el número 63 de Microhobby, omitíamos una línea en el listado sin la que no podréis completar el juego, ésta que os ofrecemosa a continuación:

```
9110 LET pu=0: LET vi=3: LET ni=
1
```

LA FAMILIA WALLY EN EL PARAISO o los héroes también lloran (II)

Por Dessas Trosso

Perdona, querido Wally, que la semana pasada te dejáramos en esta terrible encrucijada, pero la verdad es que no tenías verdaderos motivos de preocupación ya que la pecera (GOLDFISH BOWL) hará que la araña huya despavorida.

Ahora tranquilízate y coge la llave maestra (SKELTON KEY) que se halla delante de ti en esta tétrica estancia, pero ten cuidado de no soltar la pecera: la astuta araña permanece al acecho.

Vuelve hasta el templo y salta de nuevo a través del cuadro de la pared. ¡Ha llegado la hora de refrescarse! Zambúllete en el mar e intenta localizar dentro de sus cristalinas aguas un pequeño armario que en un principio encontrarás cerrado. ¡Ah! ¡Oh! ¡Buff! ¡Menos mal que has traído la llave que te permitirá desvelar sus bien guardados secretos!

¡Niiic, niiic! Media vuelta y ya está: ante tus atónitos ojos toma forma una maravillosa lata de espinacas.

Pese a tu avanzado estado de inanición, Wally, debes contenerte. ¡El futuro de tu familia depende de ella!

Ahora busca por las profundidades un enorme tapón. Abrelo y encontrarás un camino que te conduce de vuelta al templo. De allí tendrás que coger el tazón que contiene un exquisito relleno (STUFFING) y mostrárselo al gran pavo (EMU) que en tantas ocasiones te ha impedido el paso. ¡En cuanto vea el relleno se le caerán los huevos al suelo!

Aprovecha su desconcierto y cambia las espinacas por un huevo, a la vez que procuras no perder el relleno.

Compóntelas como puedas, pero el caso es que tienes que salir de allí con el huevo y la lata de espinacas.

Vete a la pantalla marcada con el rotulo «OLDE FAITHFULL». Salta a la liana de la derecha y corre tan rápido como puedas hacia el geiser que surgirá de repente. Este, con la ayuda de las espinacas, te subirá hasta un nido de águilas.

¡Si no llevaras el huevo, el aguila te materia sin remisión! Como, sin duda, todavía lo tienes, podrás hacerte fácilmente con el arco y sus correspondientes flechas que se encuentran en el nido, pero tendrás que dejar el huevo en su lugar. Vuelve a saltar a la tierra y utiliza estas primitivas armas para apartar de tu camino al molesto nativo que lleva el escudo.

Vete a donde está Wilma prisionera del indígena y dispárale un flecha...

¿A dónde habrá ido a parar esta diabólica flecha?

¿Llegará a tiempo de rescatar al simpático Herbert antes de que la sopa esté en su punto...?

¿Por qué los progamadores de Microgen tienen una mente tan retorcida...?

Las respuestas a estas incógnitas y muchas más, las encontraréis en el próximo capítulo de este lacrimógeno serial.

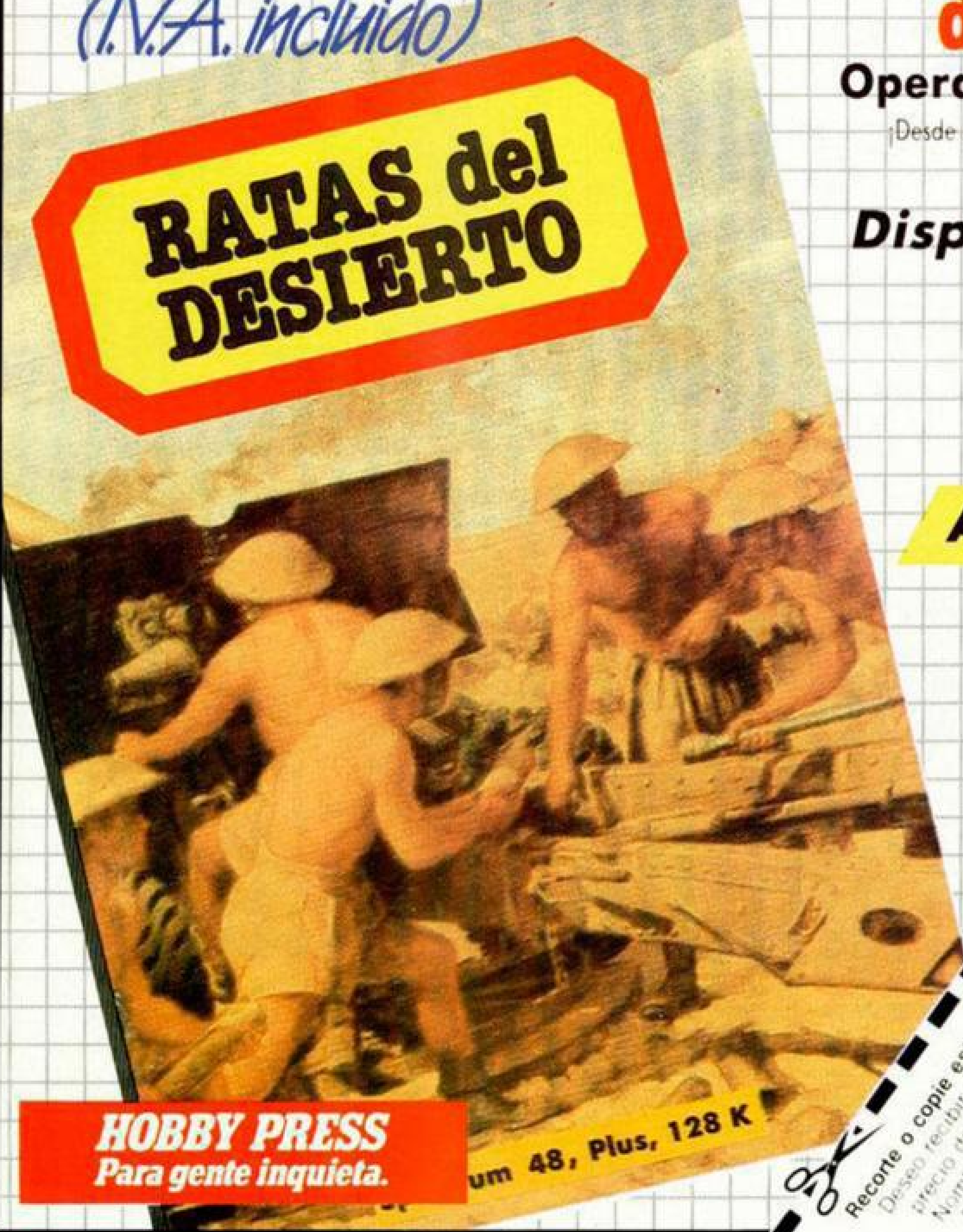
&

¡Juegos de Simulación Estratégica para poner a prueba su inteligencia y vivir la emoción de situaciones reales!

¿Se atreve Ud. a dirigir la Campaña del Desierto y derrotar a Rommel antes de que lo hiciera el General Montgomery en el Alamein?

Un **WAR GAME** de estrategia que reúne todos los condimentos necesarios para hacer de él un auténtico *«plato fuerte»*, inteligencia, emoción, sorpresa y realismo.

Sólo 1.125 pts. (I.V.A. incluido)



RATAS del DESIERTO

Operación Norte de Africa

¡Desde uno a tres jugadores!

Disponible para

Spectrum

y

Amstrad

HOBBY PRESS
Para gente inquieta.

Recorte o copie este cupon y envielo a Hobby Press, S. A. Apdo. de Correos 54.062 Madrid.

Deseo recibir en mi domicilio, **sin gastos de envio alguno por mi parte,** la cinta **RATAS DEL DESIERTO** al precio de 1.125 ptas. (IVA incluido)

Nombre

Dirección

Localidad — Provincia

Codigo — Telefono — Edad

El modelo que elijo es para

Amstrad — Spectrum

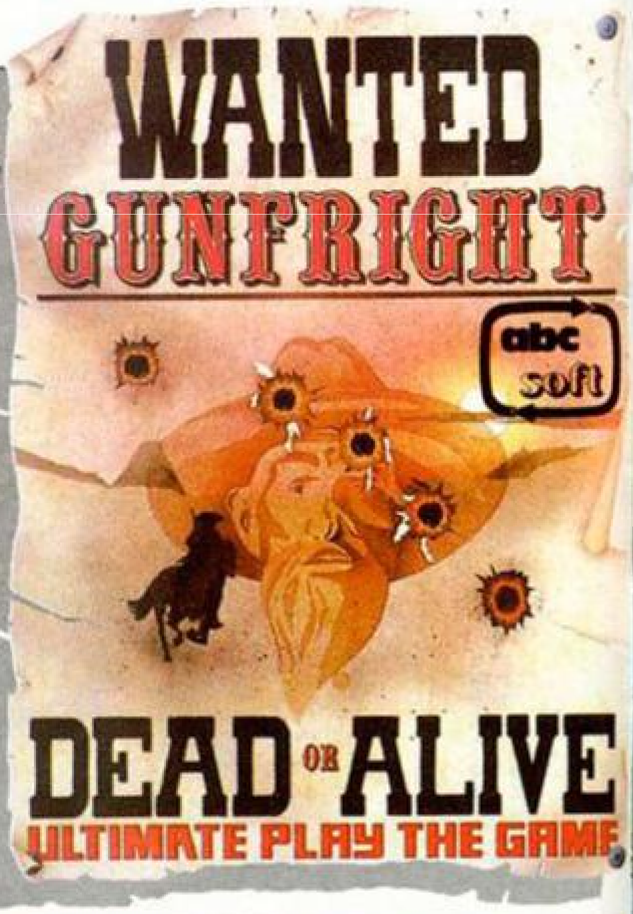
La forma de pago elegida es la que señalo con una cruz

Giro postal n.º — Tarjeta Visa n.º — Talon nominativo a Hobby Press, S. A.

Fecha de caducidad de la tarjeta — Fecha y firma

GUNFRIGHT • Ultimate • Videoaventura

VUELTA AL LEJANO OESTE



Ultimate ha cogido al protagonista de Nightshade, le ha colocado un sombrero vaquero y dos pistolas, ha cambiado las casas de piedra por cabañas de madera y ha reemplazado los malignos espíritus por peligrosos bandoleros. Y de esta forma, como el que no quiere la cosa, ha realizado este excelente y divertido Gunfright.

Vale, de acuerdo. Ya sabemos que Gunfright es muy parecido a Nightshade, que utiliza la misma perspectiva y que tanto la presentación general como la filosofía de ambos programas son muy similares. Pero ¿y qué?

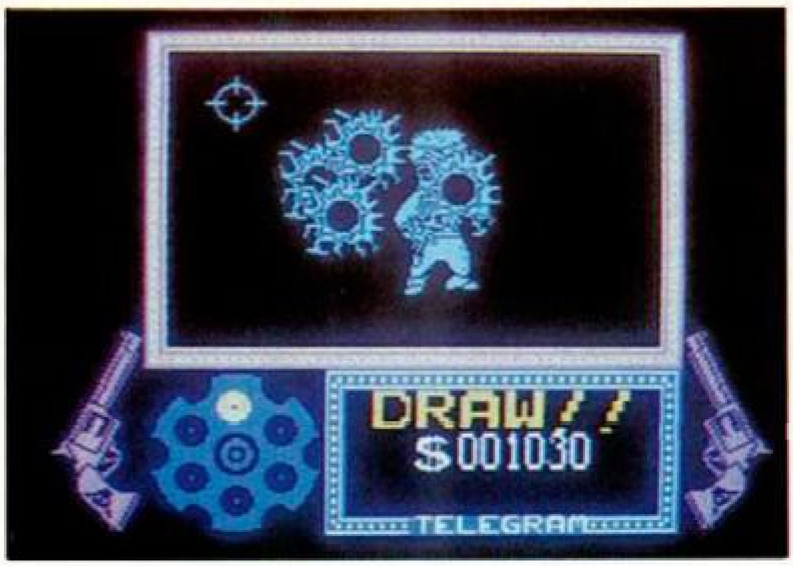
Es posible que en un principio parezca que a los excelentes programadores de Ultimate se les han agotado las ideas y que últimamente se dedican a vivir de las rentas y empiezan a repetirse en sus juegos.

Pero en cuanto empiezas a hacer tus primeras rondas por el pueblo y vas comenzando a descubrir los numerosos detalles que Gunfright posee, se te va borrando poco a poco de la cabeza toda posible relación entre ambos juegos. Te vas dando cuenta de que, en realidad, poco tienen que ver el uno con el otro y que éste es, incluso, mucho mejor y más variado que su predecesor.

Pero no seamos «racistas» y pensemos que entre nuestros lectores habrá muchas personas que no habrán ni siquiera oído hablar de Nightshade, así pues olvidémonos de prejuicios y pasemos a hablar de este juego de una manera totalmente objetiva y sin realizar ningún tipo de comparaciones, que siempre resultan odiosas.

Nos encontramos en Kinsfolk, un pequeño poblado del lejano Oeste que posee el porcentaje por metro cuadrado más alto en cuanto a forajidos y pistoleros se refiere. A nosotros nos ha tocado asumir el papel del sheriff. Evidentemente nuestra misión consiste en, fieles a nuestro juramento, hacer reinar la paz y el orden, garantizar la seguridad de todos los ciudadanos y hacer prevalecer la justicia.







Bonitas palabras, pero difíciles de cumplir. De momento, y nada más comenzar el juego, ya tendremos que estar con todos nuestros sentidos alerta, pues deberemos tratar de conseguir, haciendo uso de nuestra puntería, el mayor número posible de bolsas de dinero, las cuales nos serán imprescindibles para, durante el desarrollo de juego, poder ir comprando los objetos que necesitaremos para lograr nuestra misión, tales como munición y caballos. El precio de estos irá







variando cada cierto tiempo, por lo que además de capturar a los pistoleros, deberemos preocuparnos de hacer nuestras compras en el momento más oportuno.

Pero el objetivo primordial, como antes hemos dicho, consiste en capturar a los numerosos bandoleros que pueblan las calles de Kinsfolk. Para ello, antes, tendremos que ir enfrentándonos a digamos «pequeños malvados», que no harán otra cosa que obligarnos a gastar munición, pero que no tienen ninguna importancia para nosotros (no ofrecen ninguna recompensa por su captura).

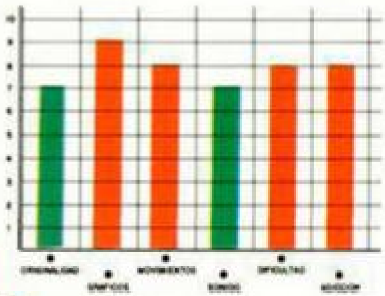
Las cosas se pondrán verdaderamente interesantes en el momento que nos encontremos con el fuera de la ley de turno, cuya fisonomía conoceremos perfectamente gracias a los carteles de «Se Busca» que se nos muestra a la izquierda de la pantalla. Aquí tendremos que enfrentarnos cara a cara con los bandoleros, llevando a cabo un peligroso duelo. No habrá clemencia por ninguna de las partes: el más rápido podrá conservar su vida, el que más tarde en reaccionar acabará mordiendo el polvo.

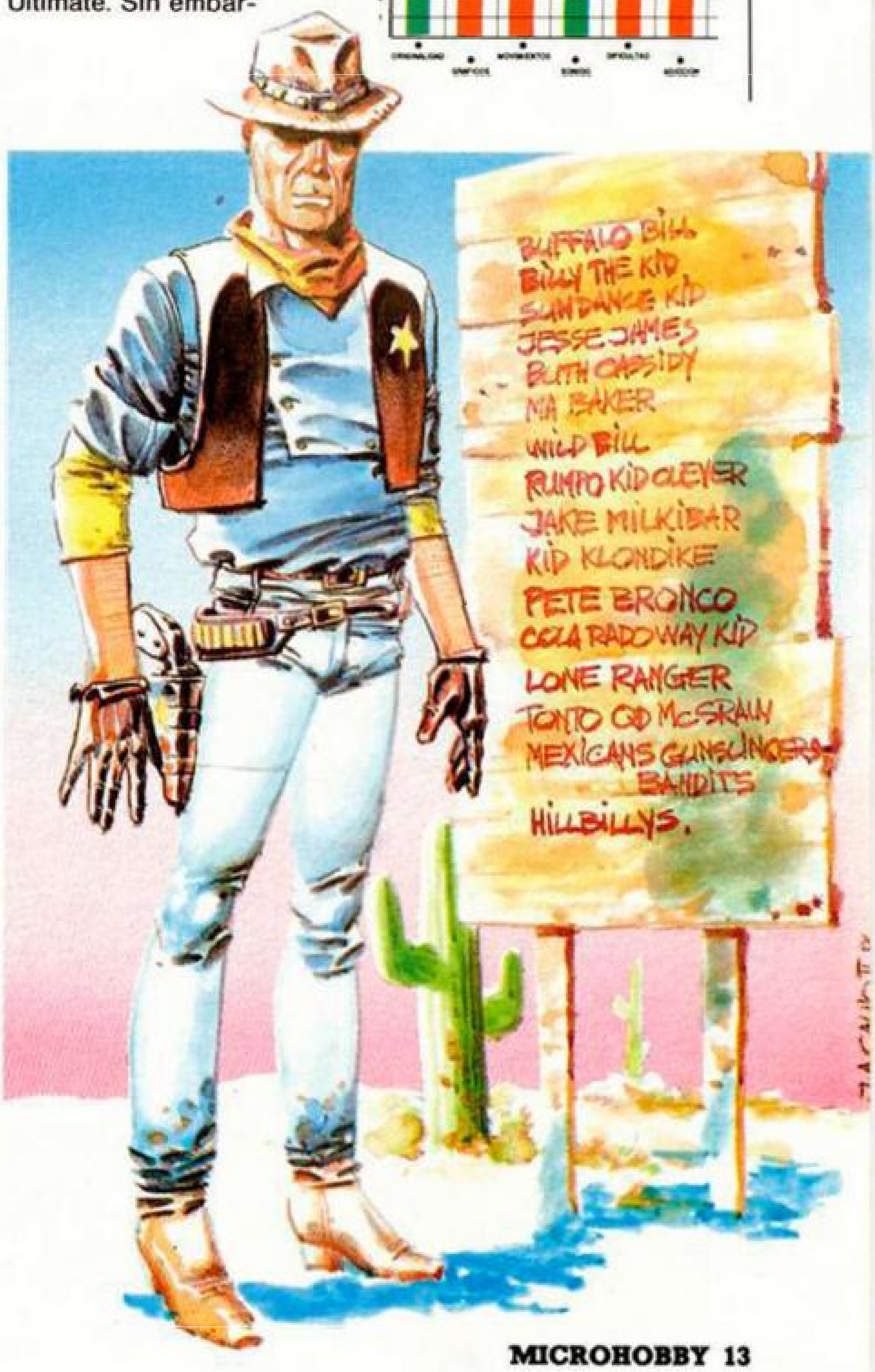
Así, tendremos que ir enfrentándonos uno a uno con un buen número de criminales. Hasta que no hayamos conseguido acabar con todos, nuestra misión no habrá concluido.

A la hora de hablar de los gráficos, si que no tenemos otro remedio que hacer mención de Nightshade, pues en Gunfright se utiliza la misma técnica, llamada Filmation 2 y que tan populares hizo a los señores de Ultimate. Sin embargo, la presentación general de la pantalla es aquí más completa ya que posee un gran número de mensajes y dibujos acompañando a las secuencias de movimiento.

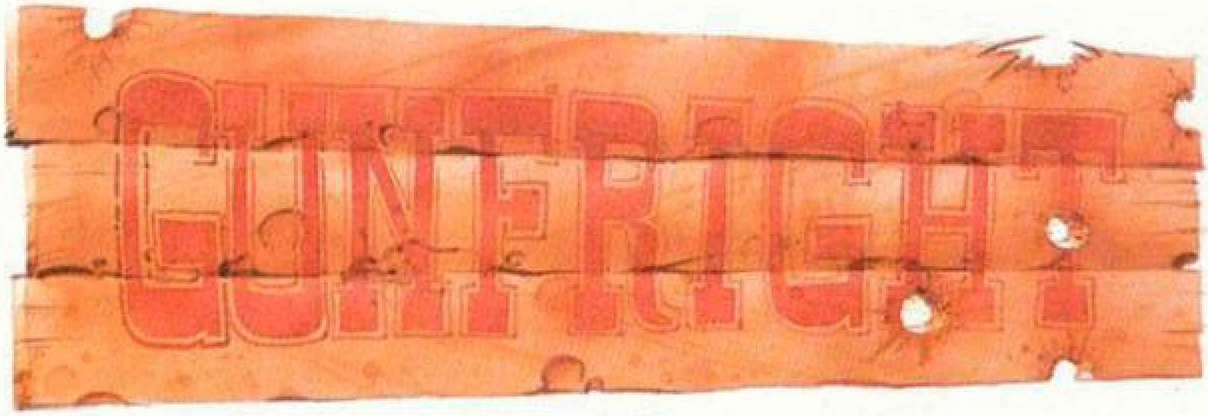
Un gran programa que nos hará olvidar el refrán de que tiempos pasados siempre fueron mejores.

Ahora, ya estáis avisados. Si os encontráis con las suficientes fuerzas y decisión, venid a Gunfrig. Vuestra recompensa será un puñado de diversión y emoción a raudales.

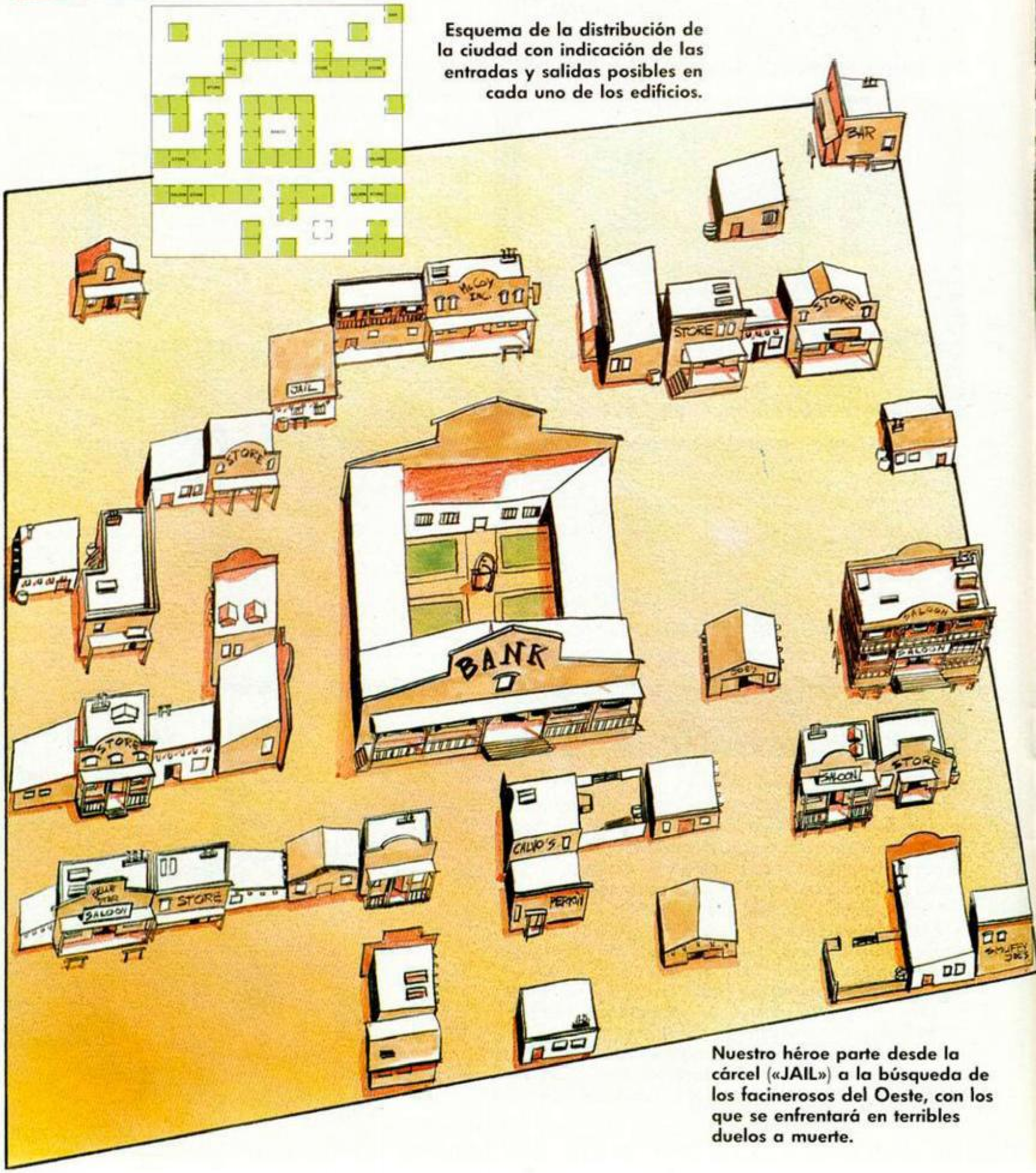




¡NUEVO!



Esquema de la distribución de la ciudad con indicación de las entradas y salidas posibles en cada uno de los edificios.



Nuestro héroe parte desde la cárcel («JAIL») a la búsqueda de los facinerosos del Oeste, con los que se enfrentará en terribles duelos a muerte.

COSMIC WARTOAD • Arcade • Ocean

EN EL IMPERIO DE LOS SAPOS

En este fascinante mundo de los juegos para ordenador, muchos y muy variados son los papeles que se nos propone representar. Podemos ser desde expertos pilotos hasta temidos pistoleros, desde hábiles karatekas a meros niños traviesos, pasando por saboteadores, jugadores de baloncesto, espadachines, astronautas y toreros. Pero hemos de reconocer que en muy contadas ocasiones nos han ofrecido la posibilidad de hacer de sapo.



Pues en Cosmic Wartoad podemos ser uno de estos viscosos y saltarines anfibios y además, podemos convertirnos en un auténtico héroe de la especie.



El caso es que la Reina de los Sapos Guerreros Cósmicos ha sido raptada por las Bestias del Fango, sus más mortales enemigos, y se la han llevado a una oscura y fría mazmorra de su castillo, donde la han transformado, con no sabemos que malignos poderes, en un horrible ser de apariencia humana. A nosotros nos ha sido encomendada la misión del rescate.

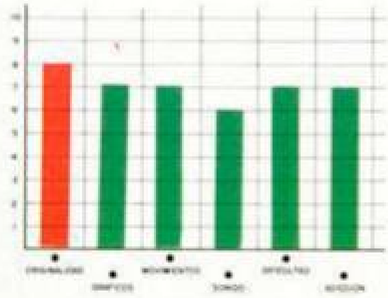
Sin embargo, Cosmic Wartoad no es el típico arcade en el que tenemos que ir recorriendo las habitaciones de una manera continuada. Entre habitación y habitación, apareceremos en un extraño tablero formado por varios cuadrados con distintos colores y dibujos. Cada uno de ellos representa una puerta hiperespacial que nos conducirá a distintos lugares, así como nos permitirá recoger uno de los objetos que aparecen a la izquierda y que te serán imprescindibles para finalizar con éxito la misión.

Este detalle le da una gran variedad al juego, ya que no tendremos que visitar siempre los mismos lugares y podremos elegir a nuestro antojo, la pantalla en la que queremos aparecer.

Estas difieren bastante unas de otras, tanto el tipo de enemigos con los que nos encontramos como en el modo en el que deberemos enfrentarnos a ellos. Habrá enormes gotas que se dejarán caer sobre nuestras cabezas, plagas de desagradables insectos, y un sinfín más de diferentes impedimentos a los que deberemos tratar de eludir a toda costa. Esto, evidentemente, le da una gran variedad al juego y hace que no resulte, ni mucho menos, aburrido.

Además, todas las pantallas están realizadas con un alto grado de calidad gráfica y son bastante agradables a la vista, tanto por sus colores como por sus formas.

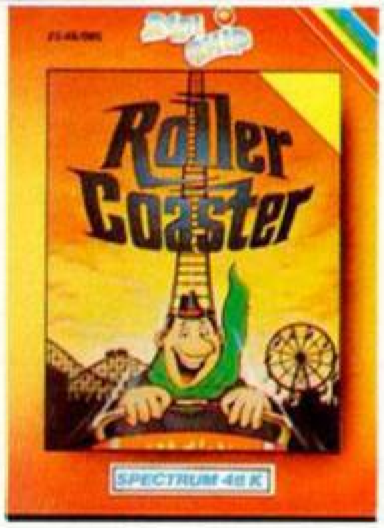
Todo ello hace de Cosmic Wartoad un juego muy ameno, variado y divertido.



ROLLER COASTER • Arcade • Elite

UN DIA DE FERIA

A primera vista Roller Coaster no llama excesivamente la atención. Quizá venga motivado por el hecho de que el tamaño del protagonista es relativamente diminuto, y así, de pronto, te deja un tanto frío.



Pero a medida que empiezas a recorrer pantallas y pantallas y vas visitando todas las instalaciones de este gigantesco parque de atracciones, vas dándote cuenta de que la primera impresión no corresponde a nada con todo lo que verdaderamente puede dar de sí este divertido arcade.

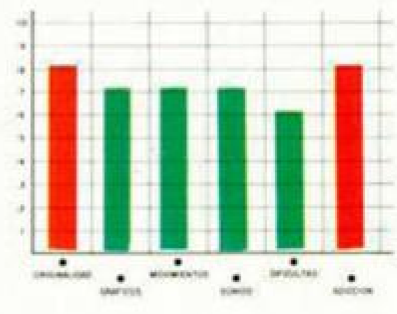
Los escenarios por los que transcurre el paseo de este personajillo en busca de diversión son realmente variados, llenos de colorido, plagados de pequeños detalles y de innumerables acciones diferentes.

El argumento consiste en recoger todas las bolsas de dinero que se encuentran por todo lo largo y ancho de esta animada feria. Para conseguirlas tendrás que ir subiéndote a las diferentes atracciones que por allí se encuentran, tales como norias voladoras, carruseles, *montañas rusas, toboganes* y un sin fin más de artefactos berbeneros.

Además, se puede apreciar en muchos detalles que este juego no está pensado con la obsesiva idea que actualmente tienen todos los programadores de que tienes que matarte cada dos por tres y que debes sudar la gota gorda para ir pasando las pantallas. Roller Coaster es un juego relativamente sencillo de jugar, hecho este



que unido a que comenzamos el juego disponiendo de diez vidas, permite que puedas pasarte un buen rato seguido jugando sin tener que volver a cada instante a comenzar desde el principio porque no hacen más que matarte insistentemente.



Para gente especialmente inquieta...



¡YA!
está a la venta el ESPECIAL n.º 2

¡Pídelo en tu kiosco!

O solicítalo directamente a nuestra editorial.

Rellena este cupón y envíalo a **HOBBY PRESS, S.A.** Apartado de Correos 232. Alcobendas. Madrid

Nombre ______________________ Apellidos ______________________

Localidad ______________________ Provincia ______________________ C. Postal __________

Teléfono ______________________ Profesión ______________________ Edad __________

DESEO RECIBIR

☐ El especial de **MICROHOBBY** n.º 2 al precio de 350 ptas. (IVA incluido)
☐ El especial de **MICROHOBBY** n.º 1 al precio de 300 ptas. (IVA incluido)
☐ Los especiales de **MICROHOBBY** nos. 1 y 2 al precio de 600 ptas. (IVA incluido)

FORMA DE PAGO

☐ Talón bancario adjungo a nombre de **HOBBY PRESS, S.A.**
☐ Contra reembolso (supone 75 ptas. de gastos de envío)
☐ Mediante Tarjeta de Crédito.
N.º ______________ Fecha de caducidad ______________

Fecha

Firma

```
(B):  01101100   6Ch
(C):  00010000   10h
```

Indicadores de condición después de la ejecución:

```
S Z   H   P/V N C
0 1 x 0 x 1   1 x
```

El octeto direccionado por "HL" contenía el mismo valor que el acumulador, por lo que el indicador "Z" se ha puesto a "1". Por otro lado, el registro "BC" permanece distinto de cero tras decrementarlo, por lo cual, el indicador "P/V" se pone también a "1".

CPIR

OBJETO:

Compara el contenido del registro acumulador con el octeto de memoria direccionado por el contenido del par de registros "HL". Si la comparación es verdadera se activará el indicador de condición "Z". A continuación incrementa en 1 el par de registros "HL" y decrementa en 1 el par de registros "BC". Si el resultado de la comparación es falso y el contenido del par de registros "BC" no es cero se repite la instrucción. La instrucción termina cuando el par de registros "BC" alcanza el valor cero o el resultado de la comparación es verdadero. Ver FIGURA 8-4.

Tenga en cuenta que si el par de registros "BC" es cero antes de la ejecución de la instrucción esta se repetirá para 64 K octetos salvo que encuentre un resultado verdadero. Esto es debido a que primero decrementa y luego compara, al decrementar 1 a 0000h, en el par de registros "BC" quedaría el valor FFFFh.

Las interrupciones no pararan la ejecución de esta instrucción por lo que se atenderían cuando terminase.

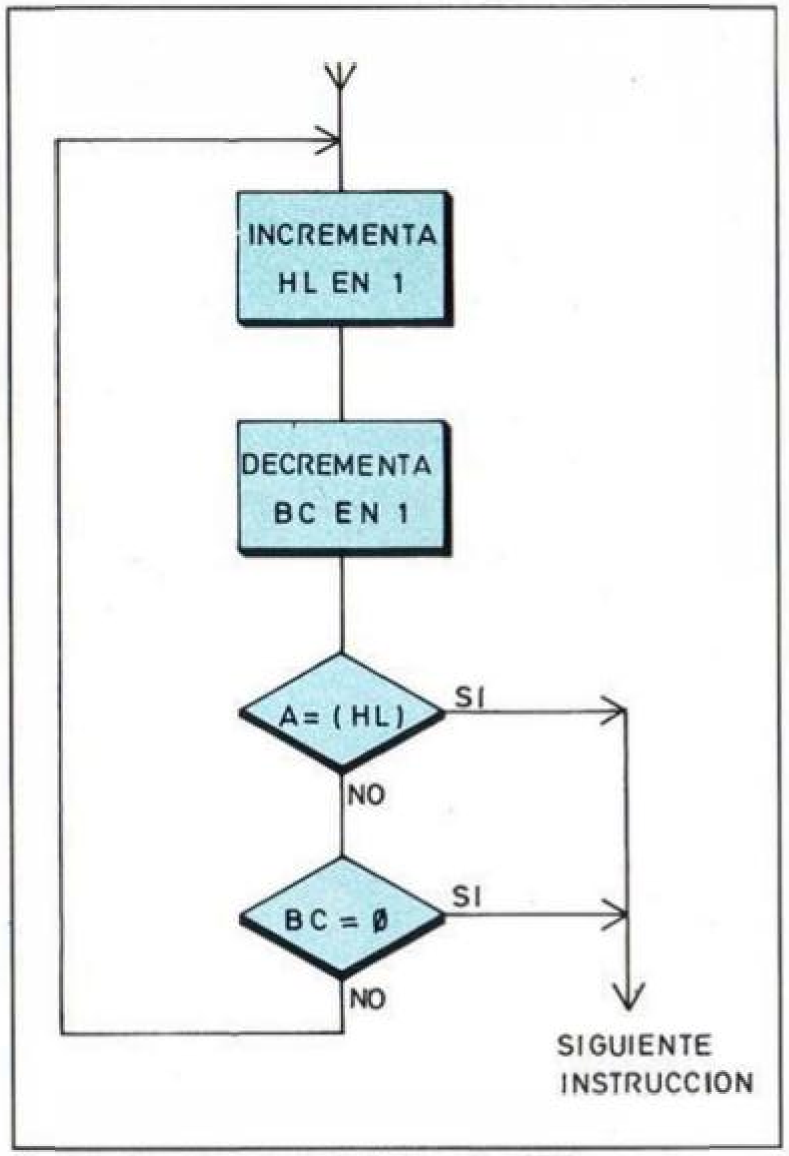


Fig. 8-4: Organigrama de la instrucción CPIR.

CODIGO DE MAQUINA:

```
11101101   EDh
10110001   B1h
```

INDICADORES DE CONDICION QUE AFECTA:

S; pone 1 - si el resultado es negativo
pone 0 - en cualquier otro caso

Z; pone 1 - si el registro "A" es igual al octeto
pone 0 - en cualquier otro caso

H; pone 1 - si no hay acarreo desde el bit 3
pone 0 - en cualquier otro caso

Grupo de instrucciones de transferencia

Supongamos que queremos transferir todo un bloque de memoria desde una zona hasta otra; por ejemplo, podría interesarnos transferir toda la pantalla a una dirección de memoria más alta (digamos, a partir de 40000) para guardar allí una copia segura.

El inicio de la pantalla es 16384 (4000h) y su longitud, con atributos incluidos, es 6912 bytes (1B00h). En Basic podríamos hacer:

```
10 LET inic=16384
20 LET dest=40000
30 LET cont=6912
40 POKE dest,PEEK inic
50 LET inic=inic+1
60 LET dest=dest+1
70 LET cont=cont-1
80 IF cont<>0 THEN GO TO 40
```

Este bucle tardaría una eternidad en ejecutarse y ocupa una cantidad ingente de memoair. Esta es una de las ocasiones en las que, o recurrimos al código máquina, o estamos perdidos. Vamos a ver cómo sería esta rutina en Assembler:

```
100          LD    HL,16384
110          LD    DE,40000
120          LD    BC,6912
130  BUCLE   LD    A,(HL)
140          LD    (DE),A
150          INC   HL
160          INC   DE
170          DEC   BC
180          LD    A,B
190          OR    C
200          JR    NZ,BUCLE
```

Esto ya es algo más racional, se ejecuta en un "abrir y cerrar de ojos" y ocupa bastante poca memoria. Hemos usado "HL" como puntero para movernos por el bloque "origen", "DE" para movernos por el "destino" y "BC" como contador de bytes a transferir. En cada pasada del bucle, hacemos la transferencia a través del registro "A", incrementamos los punteros y decrementamos el contador; si esto no es "cero", repetimos el bucle.

Es tan frecuente realizar transferencias de bloques en código máquina, que el microprocesador Z-80 posee una serie de instrucciones que nos van a ahorrar parte del trabajo en bucles de este tipo. Estas instrucciones se van a encargar de hacer todo lo que nosotros hacemos en las líneas 130 a 200; pero sin tocar el acumulador, es decir, transferir, ajustar punteros, decrementar el contador e, incluso, iterar el bucle. Son lo que se denomina: Instrucciones de transferencia.

En todas estas instrucciones el código mnemotécnico comienza por LD del inglés "load", cargar. La finalidad de estas instrucciones es transferir datos en memoria usando los pares de registros "HL" y "DE" como punteros; "HL" apuntará siempre a la primera o última dirección del bloque origen, "DE" lo mismo para el bloque destino y "BC" será el contador de bytes; como regla nemotécnica para no olvidar esto, puede asociar "DE" con la palabra "DEstino" y "BC" con "Bytes Counter" (en inglés: Contador de Bytes).

El uso más frecuente es mover campos de memoria, bien sean éstos numéricos o literales, evitando el paso por registros de CPU. El límite de longitud de estos campos lo marca la memoria disponible, teóricamente es de 64 K octetos.

El formato básico es el código mnemotécnico, que en todas ellas es distinto. Estas instrucciones no tienen operandos, ya que el direccionamiento de los mismos está implícito en el código de operación.

LDI

OBJETO:

Transfiere un octeto de memoria desde la posición direccionada por el par de registros "HL" a la posición direccionada por el par de registros "DE". A continuación incrementa 1 en ambos pares de registros y decrementa en 1 el par de registros "BC".

Equivaldría al siguiente programa:

```
100   LD    (DE),(HL)
110   INC   HL
120   INC   DE
130   DEC   BC
```

(Evidentemente, la instrucción de la línea 100 no existe, pero es una forma de ver que "LDI" transfiere el dato sin pasar por el acumulador).

CODIGO DE MAQUINA:

```
11101101   EDh
10100000   A0h
```

H; pone 0 - siempre
N; pone 0 - siempre

P/V; pone 1 - si BC-1 es diferente de cero
pone 0 - en cualquier otro caso

CICLOS DE MEMORIA:

4

CICLOS DE RELOJ:

16

EJEMPLO:

```
LDI
```

Contenido del par de registros "HL"

(H):	01010110	56h
(L):	00001010	0Ah

Contenido del par de registro "DE"

(D):	01111101	7Dh
(E):	10010010	92h

Contenido del octeto de memoria 560Ah

560Ah:	10101010	AAh

El contenido del octeto de memoria 7D92h no es significativo.

Contenido del par de registros "BC"

(B):	00000000	00h
(C):	00000101	05h

Instrucción

LDI:	11101101	EDh
	10100000	A0h

Contenido del octeto de memoria 7D92h después de la ejecución

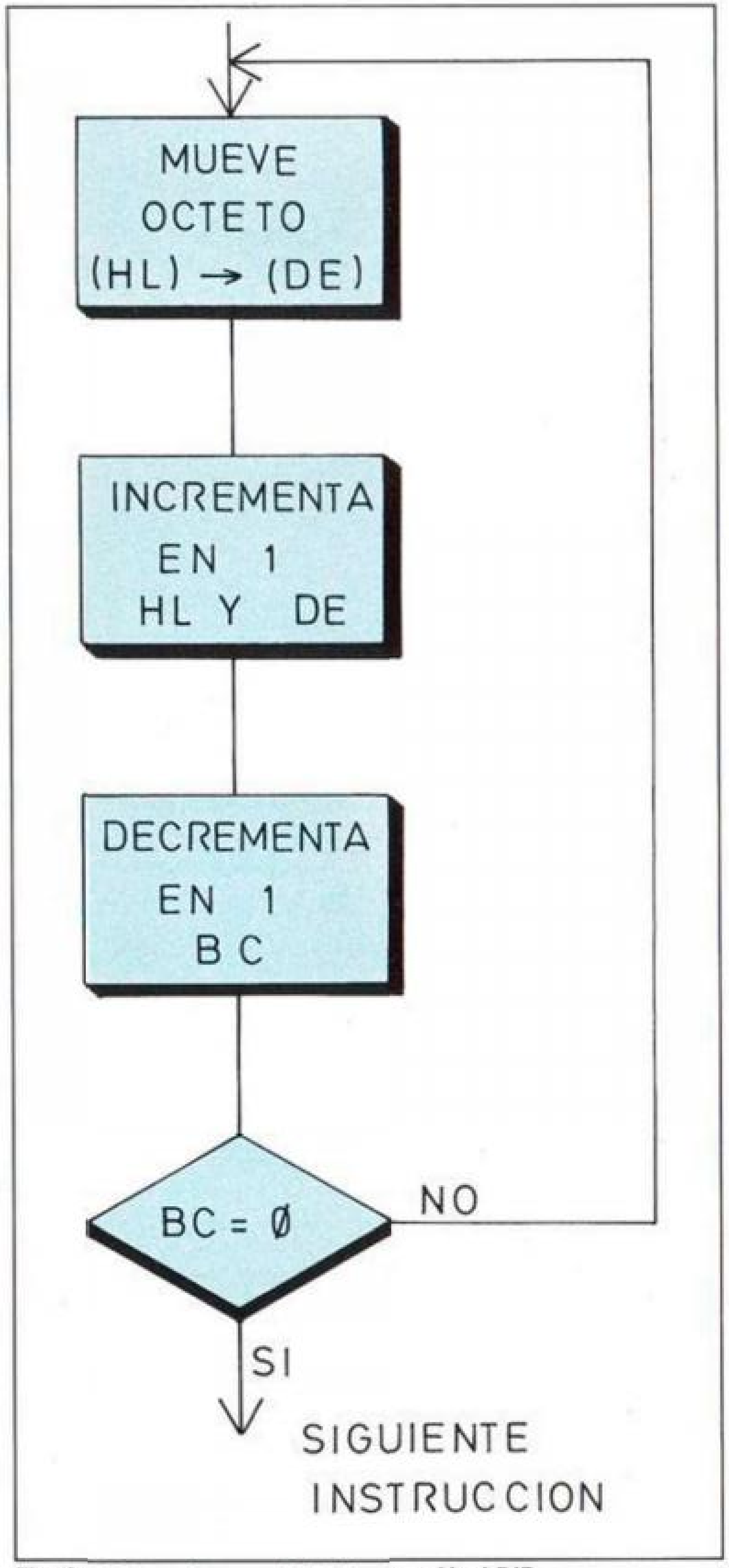


Fig. 8-2: Organigrama de la instrucción LDIR.

Lo que realmente hacen estas instrucciones, al igual que las ya vistas de comparar (CP), es una resta entre un octeto de memoria (direccionado por el contenido del par de registros "HL") y el registro acumulador, sin alterar ninguno de los dos. Por lo tanto los indicadores de condición se activarán según las reglas de las instrucciones de restar (SUB). Emplearemos a pesar de todo el término comparar, pues lo que realmente se pretende es encontrar un valor igual a uno dado y si se encuentra se activará el indicador de condición "Z" (cero).

Para más aclaraciones, si son necesarias, ver las explicaciones de las instrucciones CP y SUB.

Cuando se usan estas instrucciones se pone en el registro acumulador (A) el valor que se desea buscar.

El uso de estas instrucciones es muy variado y son de gran utilidad en el manejo de datos. Piense en lo necesario que puede ser extraer determinado dato de una tabla de gran tamaño. El inconveniente que tiene es que sólo buscan un octeto, por tanto si se quiere buscar un dato mayor es necesario empezar por encontrar uno de sus octetos y mejor es buscar el más significativo. Una vez explicadas las instrucciones analizaremos algunas técnicas de búsqueda.

CPI

OBJETO:

Compara el contenido del registro acumulador con el octeto de memoria direccionado por el contenido par de registros "HL". Si la comparación es verdadera se activará el indicador de condición "Z". A continuación incrementa 1 en el par de registros "HL" y decrementa en 1 el par de registros "BC".

Sería equivalente al siguiente programa:

```
CP  (HL)
INC HL
DEC BC
```

Con la única diferencia de que el indicador "P/V" se pondrá a "0" si "BC" alcanza un valor cero al decrementarlo y se pondrá a "1" si "BC" se mantiene distinto de cero.

CODIGO DE MAQUINA:

11101101	EDh
10100001	A1h

INDICADORES DE CONDICION QUE AFECTA:

S; pone 1 - si el resultado es negativo
pone 0 - en cualquier otro caso
Z; pone 1 - si el registro "A" es igual al octeto
pone 0 - en cualquier otro caso
H; pone 1 - si no hay acarreo desde el bit 3
pone 0 - en cualquier otro caso
N; pone 1 - siempre
P/V; pone 1 - si BC-1 es diferente de cero
pone 0 - en cualquier otro caso

CICLOS DE MEMORIA:

4

CICLOS DE RELOJ:

16

EJEMPLO:

```
CPI
```

Contenido del registro acumulador:

(A):	01110111	77h

Contenido del par de registros "HL":

(H):	10010110	96h
(L):	00110111	37h

Contenido del octeto de memoria 9637h:

9637h:	01110111	77h

Contenido del par de registros "BC":

(B):	01101100	6Ch
(C):	00010001	11h

Instrucción:

CPI:	11101101	EDh
	10100001	A1h

El contenido del registro "A" no ha variado con la ejecución.

Contenido del par de registros "HL" después de la ejecución:

(H):	10010110	96h
(L):	00111000	38h

El contenido de la posición de memoria 9636h no ha variado con la ejecución.

Contenido del par de registros "BC" después de la ejecución:

Transfiere un octeto de memoria desde la posición direccionada por el par de registros "HL" a la posición direccionada por el par de registros "DE". Después decrementa 1 los pares de registros "HL", "DE" y "BC". A continuación, compara el par de registros "BC" con cero; y si no lo es repite la instrucción. Cuando el par de registros "BC" alcanza el valor cero se pasa a ejecutar la siguiente instrucción. Ver FIGURA 8-3.

Tenga en cuenta que si el par de registros "BC" es cero antes de la ejecución de la instrucción ésta se repetirá para 64K octetos. Esto es debido a que primero decrementa y luego compara, al decrementar 1 a 0000h, en el par de registros "BC" quedaría el valor FFFFh.

Las interrupciones no paran la ejecución de esta instrucción por lo que se atenderían cuando terminase.

CODIGO DE MAQUINA:

11101101	EDh
10111000	B8h

INDICADORES DE CONDICION QUE AFECTA:

H; pone 0 - siempre
N; pone 0 - siempre
P/V; pone 0 - siempre

CICLOS DE MEMORIA:

Si "BC" diferente de 0
5

Si "BC" igual a cero
4

CICLOS DE RELOJ:

Si "BC" diferente de cero
21

Si "BC" igual a cero
16

EJEMPLO:

LDDR

Contenido del par de registros "HL"

(H):	01110100	74h
(L):	10111001	B9h

Contenido del par de registros "DE"

(D):	10000011	83h
(E):	00010111	17h

Contenido del par de registros "BC"

(B):	00000000	00h
(C):	00000101	05h

Contenido de las 5 posiciones de memoria anteriores a 74B9h

	01110111	77h
	01100110	66h
	01010101	55h
	01000100	44h
74B9h	00110011	33h

El contenido de las 5 posiciones de memoria anteriores a 8317h no es significativo.

Instrucción

LDDR:	11101101	EDh
	10111000	B8h

Contenido del par de registros "HL" después de la ejecución

(H):	01110100	74h
(L):	10110100	B4h

Contenido del par de registros "DE" después de la ejecución:

(D):	10000011	83h
(E):	00010010	12h

Contenido del par de registros "BC" después de la ejecución:

(B):	00000000	00h
(C):	00000000	00h

Contenido de las 5 posiciones de memoria anteriores a 8317h después de la ejecución:

01110111	77h
01100110	66h
01010101	55h
11000100	44h
01110011	33h

El contenido de las 5 posiciones de memoria anteriores a 47BCh no ha variado después de la ejecución.

Indicadores de condición después de la ejecución:

S	Z		H		P/V	N	C
x	x	x	0	x	0	0	x

Es importante tener en cuenta que, de estas instrucciones, se sale siempre con el registro "BC" a cero y los registros "HL" y "DE" apuntando a la dirección siguiente o anterior a cada uno de los bloques afectados.

Grupo de instrucciones de búsqueda

Las instrucciones de búsqueda tienen por objeto buscar en una tabla o posición de memoria un valor igual a uno dado.

7D92h:	10101010	AAh

El contenido del octeto de memoria 560Ah no ha variado con la ejecución.

Contenido del par de registros "HL" después de la ejecución

(H):	01010110	56h
(L):	10010011	93h

Contenido del par de registros "DE" después de la ejecución

(D):	01111101	7Dh
(E):	10010011	93h

Contenido del par de registros "BC" después de la ejecución

(B):	00000000	00h
(C):	00000100	04h

Indicados de condición después de la ejecución

S	Z		H		P/V	N	C
x	x	x	0	x	1	0	x

LDIR

OBJETO:

Transfiere un octeto de memoria desde la posición direccionada por el par de registros "HL" a la posición direccionada por el par de registros "DE". Después incrementa 1 en ambos pares de registros y decrementa en 1 el par de registros "BC". A continuación, comprueba si el par de registros "BC" vale cero; y si no, repite la instrucción. Cuando el par de registros "BC" alcanza el valor cero se pasa a ejecutar la siguiente instrucción. Ver FIGURA 8-2.

En resumen, hace lo mismo que "LDI", pero cierra el bucle mientras "BC" sea distinto de cero.

Tenga en cuenta que, si el par de registros "BC" es cero antes de la ejecución de la instrucción, ésta se repetirá para 64 K (65536 ó 10000) octetos. Esto es debido a que primero decrementa y luego compara, al decrementar 1 a 0000h, en el par de registros "BC" quedaría el valor FFFFh.

Las interrupciones no paran la ejecución de esta instrucción por lo que se atenderán cuando termine.

CODIGO DE MAQUINA:

11101101	EDh
10110000	B0h

INDICADORES DE CONDICION QUE AFECTA:

H; pone 0 - siempre
N; pone 0 - siempre
P/V; pone 0 - siempre

CICLOS DE MEMORIA (por cada octeto transferido):

Si "BC" diferente de 0
5
Si "BC" igual a 0
4

CICLOS DE RELOJ (por cada octeto transferido):

Si "BC" diferente de cero
2
Si "BC" igual a cero
16

Observe que cada repetición gasta un determinado número de ciclos, menos la última vez, cuando "BC" es igual a cero, que gasta menos. Esto es debido a que la repetición se produce decrementando en 2 el registro contador de programa "PC".

EJEMPLO:

LDIR

Contenido del par de registros "HL"

(H):	01000111	47h
(L):	10111100	BCh

Contenido del par de registros "DE"

(D):	10001000	88h
(E):	01100011	63h

Contenido del par de registros "BC"

(B):	00000000	00h
(C):	00000111	07h

Contenido de las 7 posiciones de memoria a partir de 47BCh

47BCh:	00000001	01h
	00000010	02h
	00000011	03h
	00000100	04h
	00000101	05h
	00000110	06h
	00000111	07h

El contenido de las 7 posiciones de memoria a partir de 8863h no es significativo.

Instrucción

LDIR:	11101101	EDh
	10110000	B0h

Contenido del par de registros "HL" después de la ejecución

(H):	ØlØØØlll	47h
(L):	llØØØØll	C3h

Contenido del par de registros "DE" después de la ejecución

(D):	lØØØlØØØ	88h
(E):	ØllØlØlØ	6Ah

Contenido del par de registros "BC" después de la ejecución

(B):	ØØØØØØØØ	ØØh
(C):	ØØØØØØØØ	ØØh

Contenido de las 7 posiciones de memoria a partir de 8863h después de la ejecución

8863h:	ØØØØØØØl	Ølh
	ØØØØØØlØ	Ø2h
	ØØØØØØll	Ø3h
	ØØØØØlØØ	Ø4h
	ØØØØØlØl	Ø5h
	ØØØØØllØ	Ø6h
	ØØØØØlll	Ø7h

El contenido de las 7 posiciones de memoria a partir de 47BCh no ha variado después de la ejecución.

Indicadores de condición después de la ejecución

S	Z		H		P/V	N	C
x	x	x	Ø	x	Ø	Ø	x

El mnemónico "LDI" es abreviatura del inglés: "LoaD and Increment" (Carga e incrementa). "LDIR" es abreviatura de: "LoaD, Increment and Repeat" (Carga, incrementa y repite).

Estas instrucciones permiten realizar transferencias de bloques, moviendo los punteros desde el inicio del bloque al final, es decir, desde la dirección más baja de cada bloque, hacia la más alta. Esto funciona perfectamente; incluso, si los dos bloques se solapan (tienen algunas direcciones en común), siempre y cuando, la dirección inicial del destino sea más alta que la del origen.

Pero ¿qué pasa si los bloques se solapan y el destino está más bajo que el origen? En este caso, el proceso de transferencia corrompería el bloque transferido (si lo representa gráficamente, lo verá con claridad). En este caso, sería útil disponer de instrucciones que hicieran lo mismo que "LDI" y "LDIR", pero moviendo los punteros desde el final de cada bloque hacia el principio, es decir, decrementándolos.

Como en el Z-8Ø todo está previsto, disponemos de estas instrucciones; se denominan: "LDD" (LoaD and Decrement) y "LDDR" (LoaD, Decrement and Repeat).

LDD

OBJETO:

Transfiere un octeto de memoria desde la posición direccionada por el par de registros "HL" a la posición direccionada por el par de registros "DE". A continuación decrementa 1 los pares de registros "HL", "DE" y "BC".

CODIGO DE MAQUINA:

lllØllØl	EDh
lØlØlØØØ	A8h

INDICADORES DE CONDICION QUE AFECTA:

H; pone Ø - siempre
N; pone Ø - siempre
P/V; pone 1 - si BC-1 es diferente de cero
pone Ø - en cualquier otro caso

CICLOS DE MEMORIA:

4

CICLOS DE RELOJ:

16

EJEMPLO:

LDD

Contenido del par de registros "HL"

(H):	lØØØØØØl	81h
(L):	lØlØØllØ	A6h

Contenido del par de registros "DE"

(D):	lØØlØØlØ	92h
(E):	ØØØØØØlØ	Ø2h

Contenido del par de registros "BC"

(B):	ØØllØllØ	36h
(C):	llØØlØØl	C9h

Contenido del octeto de memoria 81A6h

81A6h:	llllllll	FFh

El contenido del octeto de memoria 92Ø2h no es significativo.

Instrucción

LDD:	lllØllØl	EDh
	lØlØlØØØ	A8h

Contenido del octeto de memoria 92Ø2h después de la ejecución

92Ø2h:	llllllll	FFh

El contenido del octeto de memoria 81A6h no ha variado con la ejecución.

Contenido del par de registros "HL" después de la ejecución

(H):	lØØØØØØl	81h
(L):	lØlØØlØl	A5h

Contenido del par de registros "DE" después de la ejecución

(D):	lØØlØØlØ	92h
(E):	ØØØØØØØl	Ølh

Contenido del par de registros "BC" después de la ejecución

(B):	ØØllØllØ	36h
(C):	llØØlØØØ	C8h

Indicadores de condición después de la ejecución

S	Z		H		P/V	N	C
x	x	x	Ø	x	1	Ø	x

LDDR

OBJETO:

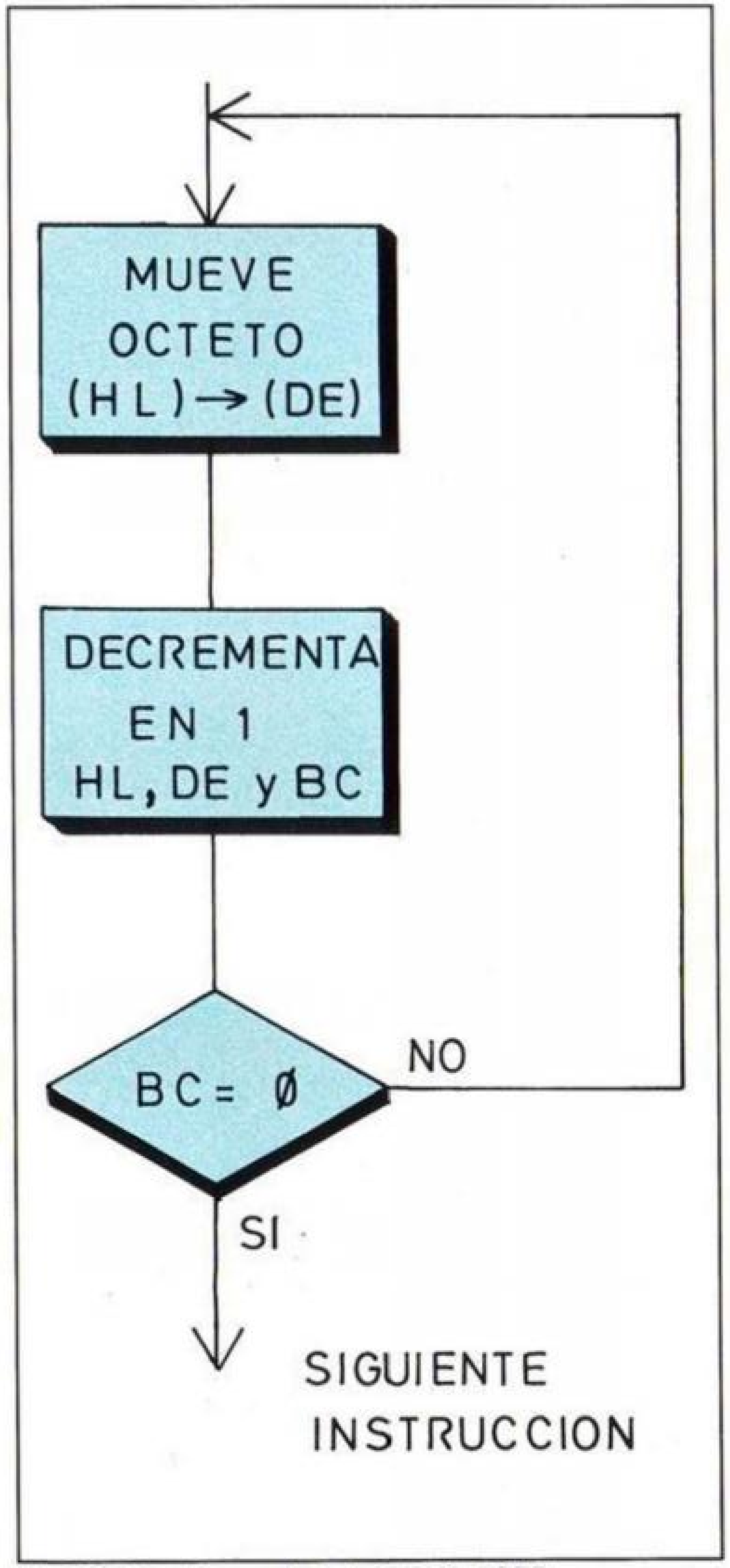


Fig. 8-3: Organigrama de la instrucción LDDR.

Profesor particular



Arturo LOBO y J. J. LEON

SISTEMAS NO LINEALES

En esta ocasión presentamos un programa dirigido a los estudiantes universitarios particularmente. Sirve para resolver todo tipo de sistemas de ecuaciones (lineales o no) hasta un total de 2Ø, con la condición de que éste no sea incompatible ni indeterminado. El programa ofrece, además, la posibilidad, si se desea, de sustituir unas ecs. por otras alternativas, hasta un total de 6, si comprueba, al hallar las soluciones, que una cadena pr que se introduce es mayor que un cierto valor real, que también se pide. Las ecuaciones son de la forma F=Ø (donde F es la cadena que se introduce) y son formas de:

a) las incógnitas, que son obligatoriamente y en ese orden las letras q, w, e, r, t, etc. de las dos primeras filas del teclado (de forma que si hay tres incógnitas éstas deben ser q, w, y e; por ejemplo),

b) vi, que es una variable independiente (puede aparecer o no) que ahora te explicaremos para qué sirve,

c) par, que es un parámetro que toma dos valores (que te pide).

Si decides introducir vi y par en las ecuaciones, puedes utilizar la opción de representación: cuando ha resuelto las ecuaciones para un valor inicial de vi y el 1.º de los dos del parámetro, te permite representar para los dos valores del parámetro, y en función de vi, una expresión de cadena cualquiera vdep. (Ejemplo: "2 w par/vi".) Te pide para ello el incremento de vi, el número de puntos y el valor máximo que puede tomar vdep.

Para resolver el sistema, debes meter siempre la primera vez valores iniciales xi para la interación (lo más próximos que tu creas a los valores solución), así como el error permitido en el resultado ei y el incremento para estimar la derivada di. El programa utiliza el método de Newton, de modo que si el valor del jacobiano en un punto de la iteración es cero, deberás introducir nuevos valores iniciales.

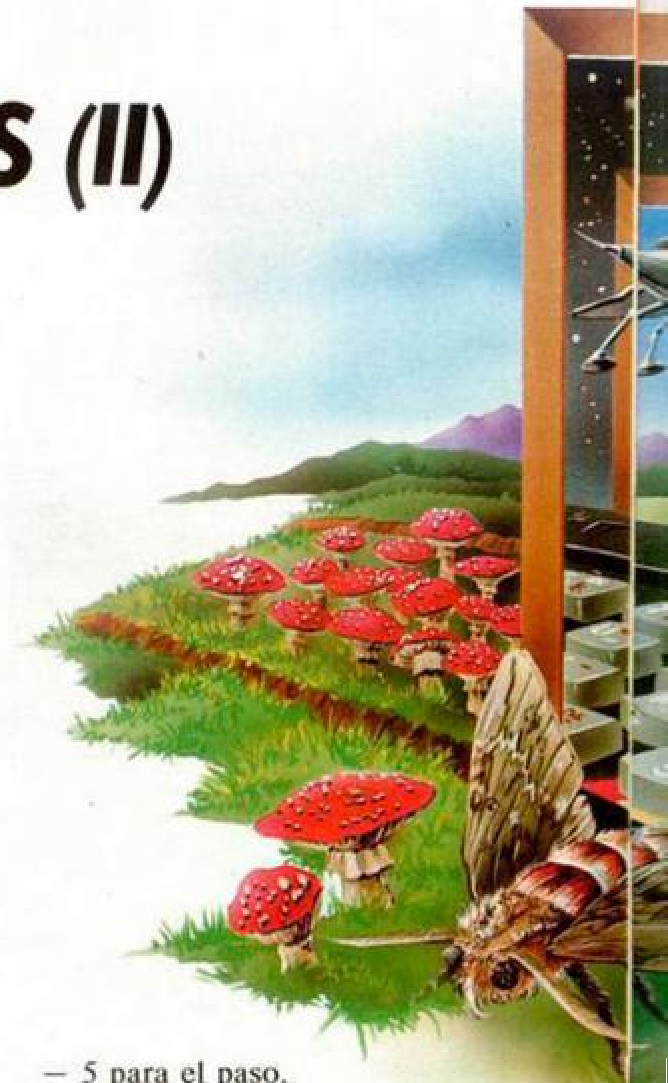
El número de ecuaciones es n y las ecs. fijas (no alternativas) deben ser introducidas en primer lugar.

TODAS LAS LETRAS MAYUSCULAS SUBRAYADAS DEBERAN TECLEARSE EN MODO GRAFICO

```
   2 INPUT "dame la extension de la cadena max",pao: LET o=1: LET cond=0: LET cu=0
   4 INPUT "si quieres tener la posibilidad de usar ecuaciones alternativas oprime 1, si no, 0",elec
   5 IF elec=0 THEN GO TO 10
   6 INPUT "numero de ecuaciones fijas,nombre de pr,valor de rel",top,a$,rel
  10 DIM r(20): DIM f(20): DIM s(20)
  11 DEF FN a(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(1))
  12 DEF FN b(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(2))
  13 DEF FN c(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(3))
  14 DEF FN d(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(4))
  15 DEF FN e(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(5))
  16 DEF FN f(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(6))
  17 DEF FN g(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(7))
  18 DEF FN h(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(8))
  19 DEF FN i(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(9))
  20 DEF FN j(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(10))
  21 DEF FN k(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(11))
  22 DEF FN l(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(12))
  23 DEF FN m(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(13))
  24 DEF FN n(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(14))
  25 DEF FN o(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(15))
  26 DEF FN p(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(16))
  27 DEF FN q(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(17))
  28 DEF FN r(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(18))
  29 DEF FN s(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(19))
  30 DEF FN t(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(20))
  31 DEF FN u(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(top+2))
  32 DEF FN v(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(top+3))
  33 DEF FN w(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(top+4))
  34 DEF FN x(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(top+5))
  35 DEF FN y(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(top+6))
  36 DEF FN z(q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z)=VAL (t$(top+7))
  37 INPUT "valores del parametro y el inicial de vi,n",par1,par2,vi,n
  49 LET cont=0: LET cuen=0: LET par=par1
  50 LET vars=(PEEK 23627)+256*(PEEK 23628)
  51 LET vio=vi: DIM t$(n+1,pao)
 112 FOR i=1 TO n+elec
 113 INPUT "Introduce la expresion de la    ecuacion ";(i),t$(i)
 114 NEXT i
 115 DIM y(n,n+1): INPUT "quieres meter valores iniciales",h$
 118 IF h$="no" THEN GO TO 170
 119 IF h$<>"si" THEN GO TO 115
 120 DIM e(20): DIM d(20): DIM x(20)
 130 FOR i=1 TO n
 150 INPUT "dame e";(i);",x";(i);",d";(i),e(i),x(i),d(i)
 160 NEXT i
 170 FOR i=1 TO n
 173 IF elec=0 THEN GO TO 180
 175 IF cond=1 AND i>top THEN GO TO 185
 180 POKE (vars-1471),(96+i): POKE (vars-1005),(96+i): POKE (vars-733),(96+i)
 183 GO TO 200
 185 POKE (vars-1471),(116+i-top): POKE (vars-1005),(116+i-top): POKE (vars-733),(116+i-top)
 188 LET cont=1
 200 FOR j=1 TO n
 210 LET f(j)=d(j)
 220 GO SUB 9997
 230 NEXT j
 240 NEXT i
 250 GO SUB 5000
 260 FOR i=1 TO n
 263 IF elec=0 THEN GO TO 270
 265 IF cond=1 AND i>top THEN GO TO 275
 270 POKE (vars-489),(96+i)
 273 GO TO 276
 275 POKE (vars-489),(116+i-top)
 280 GO SUB 9998
 285 GO TO 300
 290 NEXT i
 300 FOR i=1 TO n
 310 LET x(i)=s(i)+x(i)
 320 NEXT i
 330 GO TO 170
5000 FOR q=1 TO n-1
5010 LET j=q
5020 IF y(j,q)<>0 THEN GO TO 5050
5025 IF j>=n THEN GO TO 5150
5030 LET j=j+1
5040 GO TO 5020
5060 FOR l=1 TO n+1
5070 LET y(q,l)=y(q,l)+y(j,l)
5080 NEXT l
5090 FOR h=q+1 TO n
5100 IF y(h,q)=0 THEN GO TO 5140
5102 LET a=y(h,q)
5110 FOR i=1 TO n+1
5120 LET y(h,i)=(y(h,i)*y(q,q)/a)-y(q,i)
5130 NEXT i
5140 NEXT h
5150 NEXT q
5160 FOR q=0 TO n-2
5170 LET j=q
5180 IF j>=n THEN GO TO 5300
5185 IF y(n-j,n-q)<>0 THEN GO TO 5210
5190 LET j=j+1
5200 GO TO 5180
5210 FOR l=1 TO n+1
5220 LET y(n-q,l)=y(n-q,l)+y(n-j,l)
5230 NEXT l
5240 FOR h=q+1 TO n-1
5250 IF y(n-h,n-q)=0 THEN GO TO 5290
5251 LET b=y(n-h,n-q)
5260 FOR i=1 TO n+1
5270 LET y(n-h,i)=(y(n-h,i)*y(n-q,n-q)/b)-y(n-q,i)
5280 NEXT i
5290 NEXT h
5300 NEXT q
5320 FOR p=1 TO n
5321 IF y(p,p)=0 THEN GO TO 9100
5330 LET s(p)=y(p,n+1)/y(p,p)
5340 NEXT p
5350 RETURN
6000 IF i<n THEN GO TO 290
6001 FOR i=1 TO n
6002 LET r(i)=x(i)+s(i)
6003 NEXT i
6004 IF elec=1 AND cont=0 THEN GO SUB 7000
6005 IF cuen>0 OR cu=1 THEN GO TO 6033
6006 FOR i=1 TO n
6010 LET r(i)=s(i)+x(i): PRINT "r(";i;")=";r(i)
6020 NEXT i
6025 PRINT "si quieres representar oprime    CONTINUE": STOP
6033 IF cuen>0 OR cu=1 THEN GO TO 6048
6034 LET cu=1: CLS
6040 PLOT 20,20: DRAW 200,0: DRAW 0,155: DRAW -200,0: DRAW 0,-155
6045 INPUT "nombre  de la dependiente,valor del incremento",w$,dv
6046 INPUT "maximo de vdep",maxi
6047 INPUT "numero de puntos",maxv: DIM v(2*maxv): DIM w(maxv*2)
6048 IF cuen<=1 THEN GO TO 6050
6049 DRAW ((testx-vi)*200/((maxv-1)*dv)),(testy-VAL (w$))*155/maxi: IF cuen>(maxv-1) THEN GO TO 6100
6050 PLOT 20+((vi-vio)*200/((maxv-1)*dv)),20+(VAL (w$)*155/maxi): LET v(o)=vi: LET w(o)=VAL (w$): LET o=o+1
6055 LET testx=vi: LET testy=VAL (w$)
6060 LET vi=vio+dv*(cuen+1)
6070 LET cuen=cuen+1
6090 GO TO 170
6100 IF par=par2 THEN GO TO 9110
6103 LET par=par2: LET cuen=0: LET vi=vio: GO TO 170
7000 LET pr=VAL (a$)
7001 IF pr>rel THEN LET cond=1
7002 IF cond=1 THEN GO TO 170
7003 RETURN
7004 STOP
9100 PRINT "mete nuevos valores iniciales": GO TO 2
9110 INPUT "continuo?",s$
9111 IF s$="no" THEN STOP
9112 GO TO 2
9997 LET y(i,j)=(FN c(x(1)+f(1),x(2)+f(2),x(3)+f(3),x(4)+f(4),x(5)+f(5),x(6)+f(6),x(7)+f(7),x(8)+f(8),x(9)+f(9),x(10)+f(10),x(11)+f(11),x(12)+f(12),x(13)+f(13),x(14)+f(14),x(15)+f(15),x(16)+f(16),x(17)+f(17),x(18)+f(18),x(19)+f(19),x(20)+f(20))-FN c(x(1),x(2),x(3),x(4),x(5),x(6),x(7),x(8),x(9),x(10),x(11),x(12),x(13),x(14),x(15),x(16),x(17),x(18),x(19),x(20)))/d(j): LET f(j)=0: LET y(i,n+1)=-FN c(x(1),x(2),x(3),x(4),x(5),x(6),x(7),x(8),x(9),x(10),x(11),x(12),x(13),x(14),x(15),x(16),x(17),x(18),x(19),x(20)): RETURN
9998 IF ABS (FN c(s(1)+x(1),s(2)+x(2),s(3)+x(3),s(4)+x(4),s(5)+x(5),s(6)+x(6),s(7)+x(7),s(8)+x(8),s(9)+x(9),s(10)+x(10),s(11)+x(11),s(12)+x(12),s(13)+x(13),s(14)+x(14),s(15)+x(15),s(16)+x(16),s(17)+x(17),s(18)+x(18),s(19)+x(19),s(20)+x(20)))<=e(i) THEN GO TO 6000
9999 RETURN
```

LA ZONA DE VARIABLES (II)

Rafael PRADES



El Spectrum almacena todas las variables con las que opera en un determinado espacio de la memoria. En esta segunda parte del análisis de la Zona de Variables os presentamos el resto de los tipos existentes: de cadena, de control de bucle, de matrices numéricas y de matrices alfanuméricas.

La semana pasada vimos cómo está organizada la Zona de Variables, así como la descripción detallada de las Variables Numéricas. Ahora vamos a tratar el resto de los tipos posibles.

Contamos también con un interesante programa del que, por razones de espacio, publicaremos detalladamente su funcionamiento en el próximo número.

Variables de cadena

El primer byte, de este tipo de variables, tiene la siguiente estructura:

BYTE	
Ø 1 Ø	X X X X X

El nombre de la cadena viene determinado por el código completo del byte:

```
nombre=CHR$ (PEEK byte)
```

Los dos siguientes bytes determinan la longitud de la cadena, el primero es que de menos peso y el segundo, lógicamente, el de mayor. Cuando una cadena se define como *vacía* (""), estos dos bytes están a "cero".

El valor de la cadena se encuentra distribuido en los siguientes bytes.

EJEMPLO:

```
LET c$="FIN"
```

67	= "C"
3	longitud
Ø	3 bytes
7Ø	= "F"
73	= "I"
78	= "N"

Variables de control de bucle FOR/NEXT

Las variables de bucle necesitan más bytes para almacenar la información correspondiente a sus límites y al paso.

La estructura del primer byte es la siguiente:

BYTE	
1 1 1	X X X X X

La obtención del nombre puede conseguirse a partir de la expresión:

```
nombre=CHR$ (PEEK byte–128)
```

Los dieciocho bytes posteriores, tienen la siguiente distribución:

– 5 para el límite inferior.
– 5 para el límite superior.
– 5 para el paso.
– 2 para el n.º de línea.
– 1 para el n.º de sentencia.

En un principio, los cinco bytes destinados para almacenar la información correspondiente al límite inferior, tienen el mismo valor que el asignado en la correspondiente sentencia BASIC; pero según se va incrementando el bucle este valor se va actualizando.

EJEMPLO:

```
300 FOR t=2 TO 100 STEP 4
```

244	–128=116 = "t"
Ø	
Ø	valor
x	límite
x	inferior
Ø	
Ø	

```
9000 REM LECTURA VARIABLES
9010 CLS
9020 PRINT "          NOMBRE          ","
 DIRECCION      ": PRINT
9030 LET vars=PEEK 23627+PEEK 23
628*256
9040 LET paso=0
9050 LET vars=vars+paso
9060 REM BUCLE DE LECTURA
9070 IF PEEK vars>=65 AND PEEK v
ars<=90 THEN GO TO 9150
9080 IF PEEK vars>=97 AND PEEK v
ars<=122 THEN GO TO 9190
9090 IF PEEK vars>=129 AND PEEK
vars<=154 THEN GO TO 9240
9100 IF PEEK vars>=161 AND PEEK
vars<=186 THEN GO TO 9280
9110 IF PEEK vars>=193 AND PEEK
vars<=218 THEN GO TO 9360
9120 IF PEEK vars>=225 AND PEEK
vars<=250 THEN GO TO 9400
9130 IF PEEK vars=128 THEN PRINT
 : PRINT "              FIN
          ": STOP
9140 PRINT : PRINT "
 ERROR                ": STOP
9150 REM CADENA
9160 PRINT CHR$ (PEEK vars)+"$",
"> ";vars
9170 LET paso=PEEK (vars+1)+PEEK
 (vars+2)*256+3
9180 GO TO 9050
9190 REM NUMERICA SIMPLE
9200 PRINT CHR$ (PEEK vars),"> "
;vars
9220 LET paso=6
9230 GO TO 9050
9240 REM MATRIZ NUMERICA
9250 PRINT CHR$ (PEEK vars-32);"
()","> ";vars
9260 LET paso=PEEK (vars+1)+PEEK
 (vars+2)*256+3
9270 GO TO 9050
9280 REM NUMERICA COMPUESTA
9290 PRINT CHR$ (PEEK vars-64);
9300 LET paso=vars
9305 LET vars=vars+1
9310 IF PEEK vars>128 THEN PRINT
 CHR$ (PEEK vars-128),"> ";paso:
 GO TO 9340
9320 PRINT CHR$ (PEEK vars);
9330 GO TO 9305
9340 LET paso=6
9350 GO TO 9050
9360 REM MATRIZ CADENA
9370 PRINT CHR$ (PEEK vars-96);"
$()","> ";vars
9380 LET paso=PEEK (vars+1)+PEEK
 (vars+2)*256+3
9390 GO TO 9050
9400 REM BUCLE
9410 PRINT CHR$ (PEEK vars-128);
"#","> ";vars
9420 LET paso=19
9430 GO TO 9050
```



En un principio tendrá el valor 2; al final del bucle, alcanzará el inmediatamente posterior al límite, 102 en nuestro caso.

Ø	
Ø	límite
1ØØ	superior
Ø	(100)
Ø	

Ø	
Ø	
4	paso
Ø	(4)
Ø	

44	n.° línea
1	44+1•256=3ØØ

2	sentencia (2–1=1)

Variables de matrices numéricas

El byte primero de una variable destinada para almacenar los datos de una matriz o *array* numérica, tiene la siguiente estructura:

BYTE	
1 Ø Ø	X X X X X

La siguiente expresión permite averiguar su nombre:

```
nombre=CHR$ (PEEK byte–32)
```

Los bytes posteriores tienen la siguiente distribución:

– 2 para la cantidad de bytes ocupados.
– 1 para el número de dimensiones.
– 2 para los valores de cada una de las dimensiones.
– 5 para los valores de cada uno de los elementos.

EJEMPLO:

```
DIM s(2,2)
LET s(1,1)=3
LET s(1,2)=–2Ø
LET s(2,1)= 2
LET s(2,2)=5
```

147	–32=115 = "s"

25	longitud
Ø	25 bytes

2	dimensiones (2)

2	valor 1.ª
Ø	dimensión (2)

2	valor 2.ª
Ø	dimensión (2)

Ø	valor
Ø	primer
3	elemento
Ø	(3)
Ø	

Ø	valor
255	segundo
236	elemento
255	(–2Ø)
Ø	

236+255•256=65516
65516–65536=–2Ø

Ø	valor
Ø	tercer
2	elemento
Ø	(2)
Ø	

Ø	valor
Ø	cuarto
5	elemento
Ø	(5)
Ø	

Variables de matrices de cadena

Las variables de matriz de cadena se almacenan de forma similar a las numéricas. Cuando se define una matriz de este tipo, el área correspondiente a los elementos se rellena con el código decimal "32" (espacio).

El primer byte tiene la siguiente estructura:

BYTE	
1 1 Ø	X X X X X

Para conocer su nombre, utilizar la siguiente expresión:

```
nombre=CHR$ (PEEK byte–96)
```

EJEMPLO:

```
DIM d$(4,5)
LET d$(1)="pepe"
LET d$(2)="juan"
```

196	–96=1ØØ = "d"

25	longitud
Ø	25 bytes

2	bidimensional

4	valor 1.ª
Ø	dimensión (4)

5	valor 2.ª
Ø	dimensión (5)

112	= "p"
1Ø1	= "e" elemento
112	= "p"
1Ø1	= "e" primero
32	= " "

1Ø6	= "j"
117	= "u" elemento
97	= "a"
11Ø	= "n" segundo
32	= " "

32	= " "
32	= " " elemento
32	= " "
32	= " " tercero
32	= " "

Y así sucesivamente.

42 CARACTERES POR LINEA PARA EL SPECTRUM

Ricardo SERRAL WIGGE



Para casi todas las aplicaciones que no sean juegos, la pantalla del Spectrum se nos queda pequeña. En este artículo presentamos un programa que aumenta la capacidad para textos de la pantalla, y que podemos utilizar fácilmente con nuestros propios programas en Basic y en Código Máquina.

La pantalla del ZX Spectrum siempre trabaja en modo de alta resolución. Para que podamos escribir textos en ella, la ROM contiene una subrutina de impresión (RST #1Ø) que escribe los caracteres pixel por pixel en una matriz de 8*8 puntos.

No podemos cambiar el formato gráfico de 256*192 pixels, pero sí es posible comprimir los caracteres utilizando otras matrices 4*8, 5*8, 6*8 y 7*8 que nos darán 64, 51, 42 y 36 caracteres por línea respectivamente. 42 caracteres por línea son un buen compromiso entre densidad y legibilidad en una pantalla de TV normal.

Para poner en marcha este proyecto hubo que abordar las siguientes dificultades:

- Diseñar un juego de caracteres nuevo que utiliza una matriz 6•8.
- Los nuevos caracteres a veces se solapan con dos antiguos. Esto ocurre con los caracteres que ocupan las posiciones TAB (4•n+1) y TAB (4•n+2) (ver figura 1).
- La subrutina debe soportar todo el juego de caracteres del Spectrum, incuidos los gráficos de bloque, que se generan a través de un pequeño algoritmo, y los gráficos definidos por el usuario.
- El programa tiene que funcionar correctamente en todos los canales de salida del Spectrum, incluido el de la impresora ZX o compatibles.
- Todos los caracteres de control del Spectrum han de funcionar como de costumbre. Esto obliga a que los atributos se tengan que imprimir dos veces si un carácter nuevo (6•8) rebasa la frontera entre dos antiguos (8•8).

El resultado de estas consideraciones ha sido una subrutina en Código Máquina, que se parece mucho a la original de la ROM y que comienza en la dirección 6361Ø, pero una vez alterados los flujos de los canales, como explicaremos más adelante. Se utiliza igual que la antigua rutina de impresión mediante cualquier variante de la sentencia PRINT del Basic o mediante la instrucción RST #1Ø del Z-8Ø con el registro A conteniendo el carácter a imprimir.

El juego de caracteres comienza en la dirección 646ØØ y ocupa 768 Bytes. Es posible tener varios juegos de caracteres simultáneamente en la memoria y realizar la conmutación entre ellos alterando la variable del sistema CHARS como acostumbramos a hacerlo con la vieja rutina de impresión.

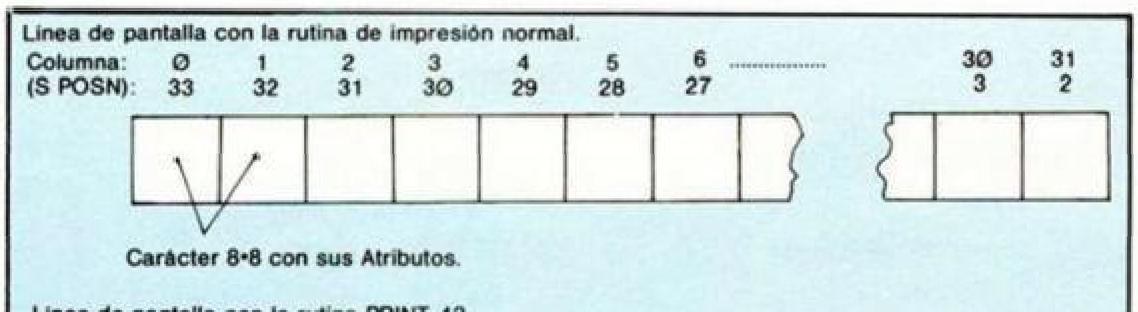


Figura 1: Formato de una línea de pantalla con PRINT-42.



Con PRINT-42 la capacidad de texto de la pantalla del Spectrum aumenta en un 31%, pero la impresión es un 33% más lenta, ya que algunos caracteres y sus atributos se han de imprimir en dos pasadas (figura 1).

Desde el Basic se trabaja exactamente igual con PRINT-42 que con la rutina de la ROM. PRINT #n, LPRINT #,n LIST #n, LLIST #n, funcionan con la misma sintaxis que de costumbre, salvo TAB y AT que ahora llegan hasta la columna 41. A veces puede ocurrir que un carácter tenga dos colores de fondo distintos, puesto que se encuentra en la frontera de dos caracteres antiguos. De todos modos, normalmente usaremos PRINT-42 para aplicaciones de textos, en las que se suele trabajar casi siempre con un mismo color de fondo. Unicamente hay que reinicializar el programa cada vez que se borra la pantalla con CLS, CLEAR o RUN, ya que estas sentencias Basic vuelven a tomar los valores iniciales para los canales «K», «S» y «P». A este efecto el programa Demostración de PRINT-42 contiene la subrutina 4000. La subrutina 5000 vuelve al modo normal de 32 caracteres por línea, permitiendo «displays» mixtos.

Examinando el programa demostración se puede ver con claridad la forma más adecuada de utilizar PRINT-42 desde el Basic. Por ejemplo vemos que, después de un CLS, se hace un GO SUB 4000 para activar PRINT-42 y cambiar el juego de caracteres. No obstante, antes de imprimir en pantalla hemos de fijar las nuevas coordenadas de referencia mediante PRINT #n; AT y, 0; o PRINT #n; TAB 0, ya que CLS las inicializa incorrectamente para la nueva rutina de impresión. Para la impresora esto se haría con LPRINT TAB 0; después de un CLEAR o RUN.

Veamos ahora cómo se puede integrar PRINT-42 en un Programa en Basic que vayamos a diseñar. Primeramente nuestro programa ha de reservar sitio para la subrutina en código máquina mediante la instrucción CLEAR 63609. Después puede cargar la subrutina con LOAD "" CODE 63610,1758 (se habrá grabado previamente utilizando el Cargador Universal de código máquina). Para que podamos activar y desactivar la rutina de impresión PRINT-42, es preciso que nuestro programa en Basic contenga las subrutinas 4000 y 5000 del programa demostración. Para quien le resulte molesto tener que incluir estas subrutinas en Basic es posible sustituirlas por dos pequeñas subrutinas equivalentes en código máquina. No obstante, es más sencilla la primera solución.

Los canales del Spectrum

Veamos ahora con detalle cómo funcionan los canales, que es una parte muy poco conocida del «sistema operativo» del Spectrum. Un canal es lo que se suele denominar en otros sistemas operativos un conjunto de «device drivers» (subrutinas controladoras de dispositivos de entrada y salida).

El Spectrum conoce 4 canales: «K», «S», «P» y «R» que controlan la pantalla inferior y el teclado, la pantalla principal, la impresora, y el WORKSPACE respectivamente. Con el interface 1 se le añaden los canales «N», «T», «B» y «M». Podemos direccionar la entrada o salida hacia un determinado canal, asociándole un flujo («stream»), que van numerados del #0 al #15. Normalmente los flujos #0, #1, #2 y #3 van unidos a los canales «K», «K», «S» y «P», pero podemos alterar estas asociaciones con la sentencia OPEN #. La clave de este mecanismo se encuentra en la zona de memoria STRMS de las variables del sistema y en la zona comprendida entre (CHANS) y (PROG). Las figuras 2 y 3 muestran su estructura con detalle.

La tabla de canales es la zona de me-

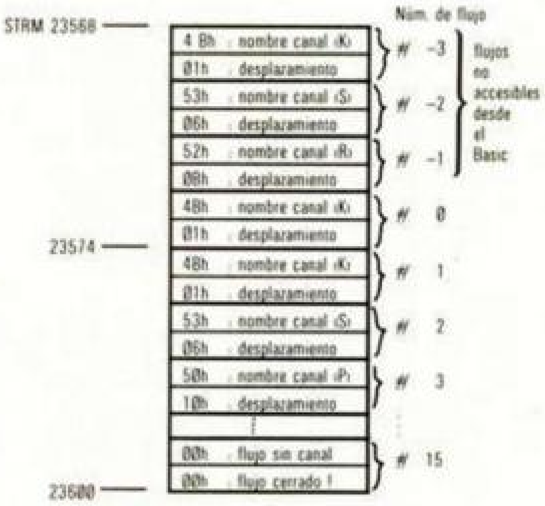


Figura 2: Tabla de flujos de las variables del sistema con sus valores iniciales.

LISTADO 1

```
  10 REM PRINT-42 DEMO
  20 REM © Ricardo Serral Wigge
  30 REM Caracteres EASAKY
  40 BORDER 1: PAPER 1: INK 9: C
LEAR 63609
  50 LOAD ""CODE 63610
  60 REM DEMO
  70 CLS : GO SUB 250: PRINT #0;
AT 0,0;: PRINT AT 0,0;
  80 PRINT PAPER 4; INK 0; FLASH
 1;"<<<"; FLASH 0;"Esto es una";
 BRIGHT 1; FLASH 1; INVERSE 1;"
demostracion "; INVERSE 0; FLASH
 0; BRIGHT 0;" "; PAPER 7; INK
1;"P"; INK 2;"R"; INK 4;"I"; INK
 5;"NT"; INK 4;"-"; INK 2;"4"; I
NK 1;"2 "; PAPER 4; INK 0; FLASH
 1;">>>"
  90 PRINT #0' PAPER 4; INK 0; F
LASH 1;"<<<"; FLASH 0;"Esto es u
na"; BRIGHT 1; FLASH 1; INVERSE
1;" demostracion "; INVERSE 0; F
LASH 0; BRIGHT 0;"  "; PAPER 7;
INK 1;"P"; INK 2;"R"; INK 4;"I";
 INK 5;"NT"; INK 4;"-"; INK 2;"4
"; INK 1;"2 "; PAPER 4; INK 0; F
LASH 1;">>>"
 100 FOR n=0 TO 41: LET a=n-10*I
NT (n/10): PRINT AT 2,n;a;AT 21,
n;a: NEXT n
 110 PRINT AT 4,0;: FOR n=32 TO
143: PRINT CHR$ n;" ";: NEXT n
 120 GO SUB 330
 130 PRINT AT 13,0;: FOR n=32 TO
 143: PRINT CHR$ n;" ";: NEXT n
 140 GO SUB 250
 150 FOR m=0 TO 10: FOR n=0 TO 4
1: LET a=n-10*INT (n/10): PRINT
OVER 1;AT 2,n;a;AT 21,n;a: NEXT
n: NEXT m
 160 CLS : GO SUB 250: PRINT AT
0,0;
 170 LIST 4000
 180 FOR n=0 TO 500: NEXT n
 190 POKE 23692,255
 200 GO SUB 330: LIST 250
 210 FOR n=0 TO 500: NEXT n
 220 GO TO 60
 230 PAUSE 0
 240 STOP
4000 REM Inicializacion
4010 LET chans=PEEK 23631+256*PE
EK 23632
4020 FOR n=chans TO chans+15 STE
P 5
4030 IF n<>chans+10 THEN POKE n,
156: POKE n+1,249
4040 NEXT n
4050 REM activar nuevo juego de
caracteres
4060 POKE 23606,88: POKE 23607,2
51
4070 RETURN
5000 REM Conectar modo normal
5010 LET chans=PEEK 23631+256*PE
EK 23632
5020 FOR n=chans TO chans+15 STE
P 5
5030 IF n<>chans+10 THEN POKE n,
244: POKE n+1,9
5040 NEXT n
5050 REM desactivar nuevo juego
de caracteres
5060 POKE 23606,0: POKE 23607,60
5070 RETURN
```

LISTADO 2

```
1   F321005B0608C5CDF40E 1041
2   C110F93E04D3FBFB2100 1270
3   5BFD7546AF47772310FC 1199
4   FDCB308E0E28C396FA00 1298
5   FDCB014EC01196FAD578 1477
6   FDCB0246C24FF9FDBE31 1542
7   3818C0FDCB02662816FD 1150
8   5E2D1D285A3E00CD0116 588
9   ED7B3F5CFDCB02A6C9CF 1547
10  04FD355220453E189032 773
11  8C5C2A8F5CE53A915CF5 1278
12  3EFDCD0116AF11F80CCD 1200
13  0A0CFDCB02EE213B5CCB 1105
14  DECBAED9CD0415D9FE20 1757
15  2845FEE22841F620FE6E 1336
16  28383EFECD0116F13291 1079
17  5CE1228F5CCDFE0DFD46 1381
18  31040E28C5CD9B0E7C0F 820
19  0F0FE603F6586711E05A 1031
20  1A4E0620EB1271132310 578
21  FAC1C9807363726F6C6C 1427
22  8FCF0CFE023880FD8631 1286
23  D619D0ED44C5472A8F5C 1297
24  E52A915CE5CD4D0D78F5 1397
25  21685C46783C7721895C 863
26  8E3803340618CD000EF1 791
27  3020E8E1FD7557E1228F 1409
28  5CED48885CFDCB0286CD 1429
29  96FAFDCB02C6C1C90000 1450
30  CD030BFE20D281FAFE06 1354
31  3869FE18306521B3F95F 1144
32  1600195E19E5C30308CE 810
33  57102954535237504F5F 702
34  5E5D5C585A54530C3E2C 745
35  892011FDCB014E200904 814
36  0E023E18862003050E28 383
37  C396FA3A915CF5FD3657 1529
38  013E20CD10F8F132915C 1095
39  C9FDCB014EC27AF80E28 1357
40  CDA2F805C396FACD0308 1434
41  793D3DE610185A3E3F18 752
42  6C112FFA320F5C180811 631
43  15FA1803112FFA320E5C 768
44  2A515C732372C9119CF9 1102
45  CD800A2A0E5C577DFE16 979
46  DA11222029444A3E2991 732
47  380CC6024FFDCB014E20 914
48  163E16900A9F1E3C4704 792
49  FDCB0246C2A2F8FDBE31 1624
50  DA860CC396FA7CCD0308 1302
51  813DC341FC57FDCB01C6 1444
52  3E20CD380C1520F8C9CD 1077
53  CCFAC3DC0A79FE163E16 1360
54  3001AFC368FA00000000 776
55  210058FDCB014E201278 829
56  FDCB02462805FD8631D6 1223
57  18C547CD9B0EC13E2891 1109
58  5F8383CB3FCB3F5F1600 1006
59  19C3DC0A865CC9FD4E45 1277
60  2A805CC9FE003840FE90 1363
61  302947CDE0FACD030B11 1075
62  925C184A21925CCDE6FA 1292
63  CB189FE60F4FCB189FE6 1326
64  F0B10E0417E6FC77230D 1107
65  20F8C9D6A53009C615C5 1336
66  ED4B785C180BCD100CC3 990
67  030BC5ED4B365CEB213B 996
68  5CCB86FE202002CBC626 1188
69  006F29292909C1EB793D 853
70  3E28200E054FFDCB014E 770
71  2806D5CD7AF8D179B9D5 1562
72  CCA2F818050000000000 643
73  D1C579E60332815CE53A 1318
74  915C06FF1F3801041F1F 652
75  9F4F3E08A7FDCB014E28 1050
76  05FDCB30CE37EB083A81 1200
77  5CFE00287DFE012845FE 1129
78  02280F3E03B0473EFCA1 844
79  4F1AA0AEA912187AC53E 1031
80  FCB0473E03A14F1AA046 1060
81  CB00CB00F53E03A047F1 1188
82  A8A912C1C5133E0FB047 1088
83  3EF0A14F1AA046CB20CB 1236
84  20A8A912C1181848C53E 962
85  F0B0473E0FA14F1AA046 1060
86  CB38CB38CB38CB38A8A9 1373
87  12C1C5133E3FB0473EC0 1053
88  A14F1AA046CB20CB20CB 1169
89  20CB20A8A912C1181814 886
90  C53EC0B0473E3FA14F1A 1089
91  A046CB38CB38A8A912C1 1296
92  08382D14233DC26FFBEB 1016
93  25FDCB014E20143A815C 903
94  FE00280AFE032806E5CD 1041
95  DB0BE12CCDDB0BE1C10D 1365
96  3A815CFE03C823C9083E 1042
97  20835F0518CC00CD282D 784
98  3E2ACD282DEF3238CD14 964
99  23CD1423FE00C3CA0A00 956
100 00000000000000001010 32
101 10101000100048480000 208
102 0000000028287C287C28 408
103 280000107C507C147C10 544
104 60640810204C0C002050 452
105 50205448340010101000 368
106 00000000081010101010 88
107 08002010101010102000 152
108 0028107C102800000000 236
109 10107C10100000000000 188
110 00000810000000007800 152
111 00000000000000303000 96
112 0000040810204000384C 256
113 54545464380010305010 568
114 10107C00384404384040 468
115 7C003844041804443800 404
116 081828487C0800007840 468
117 40780444380038404078 616
118 444438007C0404081020 380
119 20003844443844443800 472
120 3844443C040438000000 316
121 00100010000000000800 40
122 00081000000810204020 176
123 10000000000078007800 256
124 00001000080408103844 176
125 0810100010003844545C 356
126 58403C003844447C4444 664
127 44007844447844447800 700
128 38444040404438007844 628
129 4444444478007C404070 756
130 40407C007C4040704040 744
131 40003844405C44443800 536
132 4444447C444444007C10 672
133 101010107C0004040404 204
134 44443800444850605048 660
135 44004040404040407C00 576
136 446C5444444444004464 700
137 544C4444440038444444 624
138 44443800784444784040 696
139 400038444444544C3800 540
140 78444478504844003844 720
141 4038044438007C101010 420
142 10101000444444444444 456
143 38004444444444281000 452
144 44444444545438004444 632
145 28102844440044442810 424
146 101010007C4408102044 364
147 7C003C20202020203C00 404
148 00402010080400007808 252
149 08080808780010285410 308
150 10101000000000000000 48
151 007E1C20202078207C00 526
152 000038043C443C004040 376
153 40784444780000003844 564
154 404438000404043C4444 396
155 3C000000384478403800 424
156 18202070202020000000 296
157 3C44443C043840404078 628
158 44444400100030101010 316
159 38000400040404044438 200
160 4040485060504C002020 596
161 202020201C0000007854 360
162 54545400000078444444 576
163 44000000384444443800 384
164 00007844447840400000 504
165 3C44443C040400001C20 324
166 20202000000038403804 276
167 38002020782020201C00 364
168 0000444444443C000000 332
169 44444428100000004444 396
170 54543800000044281028 388
171 440000004444443C0438 392
172 00007C0810207C001C30 380
173 306030301C0010101010 332
174 101010107018180C1818 284
175 70001428000000000000 172
176 3048B4A4A4B448300000 928
```

moria apuntada por la variable del sistema CHANS. A cada canal le corresponden 5 Bytes. Inicialmente hay 4 canales. No obstante podemos incluir más canales, añadiéndolos al final de la tabla. ¡Si hacemos esto hay que tener en cuenta, que hay que desplazar «para arriba» todo el Basic un determinado número de Bytes, actualizando todas las variables del sistema afectadas por el cambio! Los dos primeros Bytes de cada canal contienen la dirección de comienzo de la subrutina que controla el dispositivo de salida, asociado al canal. Los dos siguientes contienen la dirección del dispositivo de entrada. Estas direcciones se almacenan en el formato del Z-80: primero el Byte menos significativo y después el más significativo. Si un canal es unidireccional, se toma la subrutina de ERROR-J en sustitución del dispositivo que falta, para evitar que el sistema se pueda quedar colgado. Por ello la sentencia INPUT # 2;A da ese error. Por último, el quinto Byte contiene el código ASCII del nombre del canal en mayúsculas.

Como sólo hay un número finito de flujos (16), el Spectrum tiene un espacio fijo reservado para estos datos: 32 Bytes llamados STRMS en el área de variables del sistema. A cada flujo le corresponden 2 Bytes. El primero de ellos contiene el código ASCII (¡en mayúsculas!) correspondiente al nombre del canal al que está asociado dicho flujo. Con esto sería suficiente, pero para agilizar la indexación en la tabla de canales se le añade un Byte, que contiene un desplazamiento. Con (CHANS)−1+desplazamiento obtendremos ahora la dirección de comienzo del primer Byte de los datos del canal. Con un desplazamiento cero se indica que ese flujo está cerrado.

A parte del método visto en el listado 1 hay otra forma de activar PRINT-42 y que mantiene la inicialización aún después de CLS, CLEAR o RUN. No obstante, sigue siendo necesario volver a fijar las coordenadas de referencia después de ejecutar cualquiera de estas sentencias Basic. Este método alternativo consiste en definir uno o varios canales. Definimos, por ejemplo, un canal que llamamos «A» que tome como dispositivo de entrada la rutina de lectura del teclado, y como dispositivo de salida la subrutina PRINT-42 (punto de entrada 639ØØ). Ahora, mediante la instrucción OPEN #2,«A» podemos revertir la rutina de impresión normal hacia PRINT-42.

	Campo	Valor (hex)	Nombre subrutina/canal
(CHANS)	Entrada	F4 Ø9	PRINT-OUT Ø9F4h
	Salida	A8 1Ø	KEY-INPUT 1ØA8h
	Nombre	4B	(K)
	Entrada	F4 Ø9	PRINT-OUT Ø9F4h
	Salida	C4 15	ERROR-J 15C4h
	Nombre	53	(S)
	Entrada	81 ØF	ADD-CHAR ØF81h
	Salida	C4 15	ERROR-J 15C4h
	Nombre	52	(R)
	Entrada	Ff4 Ø9	PRINT-OUT Ø9F4h
	Salida	C4 15	ERROR-J 15C4h
	Nombre	5Ø	(P)
		8Ø	Marca final de la tabla!

Figura 3: Tabla de canales del área de memoria CHANS con los canales del Spectrum sin Interface I.

ALGUNOS DATOS DE IMPORTANCIA

Esta rutina imprime 42 caracteres por línea, soportando todos los caracteres de control del ZX Spectrum. Para inicializarla y conectar el nuevo juego de caracteres 6•8: GOSUB 4ØØØ. Para volver al modo normal y desconectar el nuevo juego de caracteres: GOSUB 5ØØØ.

RUN, CLEAR & CLS reinicializan el modo de impresión de 32 caracteres por línea, pero el juego de caracteres permanece activo.

Ubicación de la rutina: 6361Ø

Número de bytes: 99Ø

Ubicación de juego de caracteres: 646ØØ

Número de bytes: 768

Número de bytes total (rutina + caracteres): 1758

Podemos obtener la dirección que ocupa un determinado carácter con la fórmula: 64344 + 8 • CODe "Caracter".

Generación del código objeto con el listado 2 y el cargador universal de Código Máquina: Dump en la 4Ø.ØØØ SAVE "print 42" CODE 4Ø.ØØØ,1758

No olvidar, en la línea 50 del listado 1, LOAD " " CODE 6361Ø.



El arte de rotular

BRADWAY SOFTWARE: LETTA-HEAD/DUMPY

Si tienes una impresora de matriz de puntos con posibilidades gráficas, estás de enhorabuena; estos dos programas son justo lo que necesitas para sacarle el máximo rendimiento.

Nos llega desde Inglaterra la producción de una pequeña empresa de software, desconocida aún en nuestro mercado, bajo la forma de dos programas de utilidad para emplear con impresoras matriciales.

En primer lugar, bajo el nombre «LETTA-HEAD» nos presentan un programa diseñador de cabeceras y etiquetas, aunque describirlo así, tal vez no dé una idea clara de las enormes posibilidades de esta excelente herramienta de trabajo.

Podríamos decir que «LETTA-HEAD» permite hacer todo lo que se pueda hacer con letras... y algo más. De entrada, y para que nunca nos quedemos cortos, el programa incluye nada menos que 25 juegos de caracteres diferentes que se pueden cargar de uno en uno o en bloques de cinco juegos. Siempre hay quien no se conforma con 25 tipos de letra diferentes; si es tu caso, la cinta incluye un «editor de juegos de caracteres» que te permitirá crearte letras a tu medida.

Una vez elegido el tipo de letra que vamos a utilizar, podemos ampliarla horizontal o verticalmente hasta 9 veces, ponerla de lado e, incluso, «boca abajo». Existe un juego que permite escribir con «imagen de espejo» y dos juegos con letra «cirílica» para escribir en griego ¡¡o en ruso!! Además, si tienes una impresora «Star Gemini», puedes volcar cualquiera de estos juegos de caracteres en la RAM de la impresora y que los utilice como los suyos.

También permite «recuadrar» una zona de la pantalla, trazar líneas, círculos o arcos para componer cabeceras de folios, membretes, logos, etiquetas, etc. Como detalle curioso, hay una opción que nos permite diseñar las etiquetas de nuestras cintas de cassette; para ello, se recuadra una zona de pantalla con la forma de una de estas etiquetas que, cuando la copiemos en impresora, tendrá el tamaño justo para recortarla y pegarla en una cassette.

Otra opción nos permite numerar automáticamente una serie de hojas, delimitando un lugar en la cabecera donde se irán imprimiendo los números consecutivos que el propio programa se encarga de ir incrementando.

Los autores han pensado en todo, y «LETTA-HEAD» es compatible con, prácticamente, todos los periféricos del mercado. El pograma principal puede trabajar con cassette, microdrive y disco «OPUS», y en la misma cinta se incluye una versión especial para «Waffadrive». Se puede indicar al programa el tipo de interface de impresora que se va a utilizar; dado que el programa es de origen inglés, no incluye en su «menú» el interface «Indescomp» (uno de los más extendidos en España), por lo que, quienes utilicen este interface, deberán seleccionar el «MOREX (centronics)» que trabaja igual.

Por último, «LETTA-HEAD» permite cargar y salvar el diseño realizado, en cualquier dispositivo de almacenamiento, como una «SCREEN$», con lo que es posible incluirlo en un programa nuestro, o mezclar los diseños de un programa de dibujo con los caracteres de «LETTA-HEAD».

El segundo de los programas, «DUMPY» es un «programa para hacer programas». Permite diseñar rutinas en código máquina para volcar pantallas en impresora, expandiéndolas, contrayéndolas, e incluso, sombreándolas para simular los colores. Consta, básicamente, de una «biblioteca» de rutinas preescritas que el programa se encarga de combinar para crear una rutina de volcado a nuestra medida. Podríamos decir que se trata de un auténtico «compilador» de rutinas de COPY.

Los cinco bloques, de cinco juegos de caracteres cada uno, están grabados por la cara «B» de la cassette de forma que podrás usarlos en tus propios programas.

La dirección de estos magníficos programadores es:

Bradway Software
33, Conalan Avenue
Sheffield
S17 4PG
ENGLAND





SPECTRUM 128
EL SUMMUM

Spectrum, como líder, marca un nuevo hito en la historia de los ordenadores familiares.

El Spectrum 128.

Gran capacidad de memoria. Teclado y mensajes en castellano, teclado independiente para operaciones numéricas y de tratamiento de textos...

Sinclair e Investrónica han desarrollado una auténtica novedad. En ningún lugar del mundo, salvo en los Distribuidores Exclusivos de Investrónica, podrás encontrar el nuevo Spectrum 128.

Sé el primero en tener lo último.

SPECTRUM 128. NOVISSIMUS

DISTRIBUIDOR EXCLUSIVO

investronica

Tomás Bretón, 62.
Tel. (91) 467 82 10.
Telex 23399 IYCO E.
28045 Madrid

Camp, 80.
Tels. (93) 211 26 58 - 211 27 54.
08022 Barcelona

LOS JUSTICIEROS DEL SOFTWARE

THREE WEEKS IN PARADISE

Wally sigue haciendo de las suyas en esta última creación de Mikro-Gen que no por recordarnos a las anteriores es peor. Pero como en esta ocasión la crítica no es cosa nuestra, cedemos la palabra a los Justicieros...

«Presentación recargada»

• **Positivo:** Los gráficos, el movimiento y el sonido. La idea del juego es bastante original, al igual que la adición. Es el mejor de la saga de «Wally».
• **Negativo:** La presentación, por estar muy cargada. Es un juego difícil ya que no consigues encontrar la entrada a nuevas pantallas.
Puntuación: 9

JUSTO SORIA.



«Gran colorido de gráficos»

• **Positivo:** El movimiento es suficientemente bueno, los gráficos tienen mucho colorido, quizás lo que hace al juego más atractivo. Es un juego muy adictivo y la curiosidad aumenta cada vez más. Hay detalles muy simpáticos y el sonido es estupendo.
• **Negativo:** La dificultad es alta porque tienes que encontrar el objeto preciso para pasar algunas pantallas que pueden ser decisivas para conseguir nuestro objetivo. No es un juego muy original porque se parece al Herbert's Dummy Rum.
Puntuación: 9

PURIFICACION HERNANDEZ.

«Un toque de humor»

• **Positivo:** Es un programa donde impera el color y la excelente calidad de sus gráficos y movimientos. El paso de pantallas es muy bueno, al principio suena una música muy bien realizada. El programa imprime una nota de humor con nuestro simpático personaje y su forma de moverse cuando se cae al suelo.
• **Negativo:** Es tremendamente difícil y hay que tener mucha paciencia para acabarlo. Es un programa hecho para quedarse muchas horas frente al ordenador.
Puntuación: 9

JAVIER REDONDO.

«Sonido perfecto»

• **Positivo:** Los gráficos y el colorido son excelentes y el movimiento, además de tener calidad, es muy gracioso. El sonido es perfecto y es un juego muy original.
• **Negativo:** La misión es muy difícil de realizar.
Puntuación: 8

DANIEL GONZALEZ.



«Excesiva dificultad»

• **Positivo:** Mantiene la originalidad de las aventuras anteriores, innovando movimientos como por ejemplo el nadar, los gráficos en su estructura son parecidos a los otros, pero con detalles muy tropicales. El movimiento es excelente y el sonido está mejorado.
• **Negativo:** Sinceramente, creo que se han pasado en dificultad.
Puntuación: 9

GORKA POLITE.

«Una idea genial»

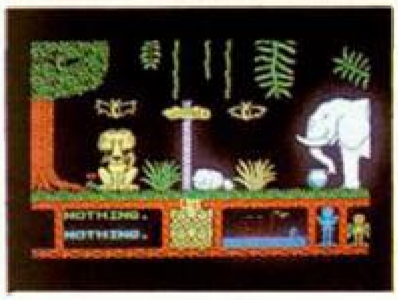
• **Positivo:** Los gráficos son maravillosos así como el movimiento, que es casi perfecto. La música del comienzo y los efectos del sonido están muy bien realizados. La idea es genial y una cosa curiosa es la de poder cambiar de color del personaje.
• **Negativo:** Los atributos se mezclan con demasiada frecuencia.
Puntuación: 9

FCO. JAVIER VALERO.

«Sigue siendo Wally...»

• **Positivo:** Los gráficos son excepcionales y el movimiento muy suave. Es un juego técnicamente perfecto. La melodía que aparece durante él es genial.
• **Negativo:** Enorme dificultad, no es muy original ya que está basado en los demás juegos de Wally.
Puntuación: 8

CELIA MENENDEZ.



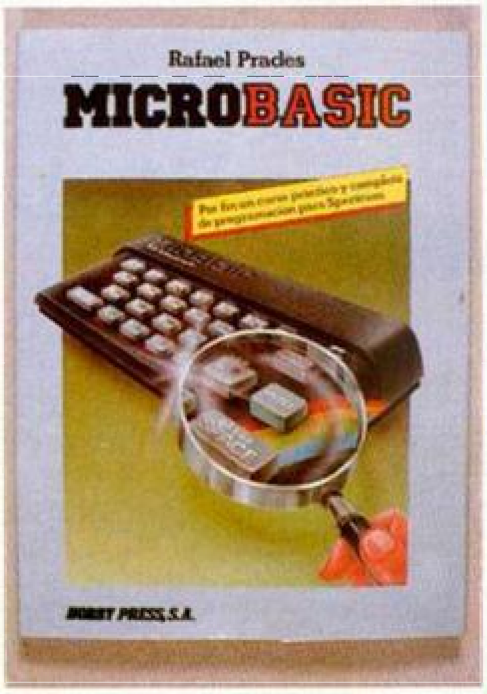
«Adición y originalidad a tope»

• **Positivo:** La pantalla de presentación y los gráficos, sin excepción son fenomenales. El tratamiento del color es excelente y no digamos el movimiento y el sonido que son de lo más llamativo. La adición, una vez metidos en el juego, y la originalidad son muy grandes.
• **Negativo:** La superposición de colores del personaje central (Wally) con el resto de los gráficos resulta algo molesto. No tiene muchas pantallas.
Puntuación: 9,5

JOSE CARLOS NUÑEZ.



LIBROS



MICROBASIC

Rafael Prades. Hobby Press, S. A.

328 páginas

Tras múltiples peticiones de los lectores de nuestra revista que, por algún motivo, no habían podido seguir asiduamente los capítulos del coleccionable «Curso de Basic», que semanalmente hemos incluido durante los números 1 al 41, hemos decidido publicar la edición completa de todos ellos en forma de libro.

La obra está enfocada a todas aquellas personas que, sin tener conocimientos previos decidan aprender de forma autodidacta el lenguaje BASIC.

Los puntos más destacables de esta obra son:

– Dedicación muy específica al BASIC/Sinclair, con aplicaciones concretas y descripción detallada de cada uno de los comandos que lo componen.

– No se necesita ningún tipo de experiencia ni conocimientos específicos previos. Los capítulos están estructurados para que el grado de dificultad y aprendizaje sean progresivos.

– Cada uno de los capítulos está profusamente ilustrado con ejemplos demostrativos y aplicaciones. Nunca se habla de conceptos que no hayan sido anteriormente explicados.

– Edición a todo color, lo que no es en absoluto habitual en este tipo de obras.

Sin ninguna duda, se trata del más completo compendio que se ha publicado sobre el BASIC de Sinclair, sus peculiaridades y aplicaciones, tanto de juegos como de utilidades.

MICRO Manía

Sólo para adictos

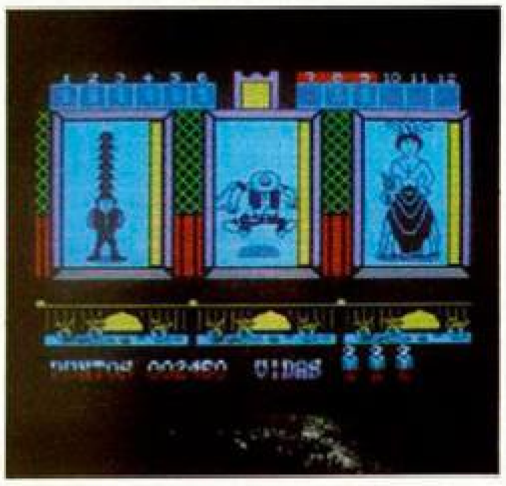
WEST BANK

Ricardo Jiménez Beteta nos envía desde Aguilar del Río un truco verdaderamente interesante y que hará las delicias de muchos aficionados a este emocionante West Bank.

Se trata de conseguir vidas infinitas, pero sin necesidad de introducir ningún cargador ni ningún Poke. Simplemente con pulsar simultáneamente las teclas:

T	V
Y	B
G	SPACE
H	Ø

Pero tener cuidado de teclearlas todas a la vez, puesto que si no pulsáis la tecla de inicio (Ø) a la vez que las demás, todos vuestros esfuerzos habrán sido en vano.



GYROSCOPE

Cuantas veces no habréis deseado que apareciera este suculento POKE en esta página. Pues aquí lo tenéis. ¡El Poke de las vidas infinitas del Gyroscope!



POKE 53922,Ø

Supongamos que ya sabréis cómo se meten los Pokes en los programas, pero por si aún no os habéis enterado, os lo repetiremos por antepenúltima vez. Teclear MERGE"" y poner en marcha la cinta. Cuando aparezca el mensaje O.K. introducir el Poke inmediatamente antes del RANDOMIZE USR. Después darle al RUN y seguir cargando la cinta.

Gracias, Oscar García Suárez (Ya creías que nos íbamos a olvidar de ti, ¿eh?).

INTERNATIONAL KARATE

Esto más que un truco es el descubrimiento público de un ligero fallo de este buen programa International Karate. Si pulsas simultáneamente las teclas W y S, te irás cargando sin ningún esfuerzo a cuantos rivales se interpongan en tu camino. Vamos, ¡que podrás estar toda la noche matando chinos!

Si esto de pulsar dos teclas te resulta demasiado complicado, coge tu joystick, ponlo hacia arriba (↑) y mantén apretado el botón de fuego. Podrás darle varias vueltas al marcador.





FLOPPY DISK INTERFACE BETA. MANEJO DESDE BASIC

Rafael PRADES

En artículos anteriores se explicó la forma en que se manejaban los ficheros, almacenados en un diskette, mediante comandos directos del D.O.S. En este otro veremos cómo incorporarlos en nuestros programas BASIC.

Cualquier comando del D.O.S. puede incluirse en una línea de programación BASIC, siempre que tenga una estructura similar a la que se muestra a continuación, por ejemplo, con el comando «CAT»:

```
20 RANDOMIZE USR
   15363:REM:CAT
```

La llamada al D.O.S. se ve ligeramente modificada, ya que en lugar de hacerlo a la dirección 15360 (3C00 Hex.), se realiza a la 15363 (3C03 Hex.).

El comando «REM» sirve para poder editar algunos comandos del D.O.S. que no son reconocidos por el intérprete del BASIC, como por ejemplo el salvar programas en código máquina con tres parámetros o los de selección de unidad; ejemplos:

```
SAVE "deep"
   40000,255,40100
•"C:"
```

ya que el sistema operativo ignora lo que hay a continuación del «REM», dentro de la misma línea.

A continuación, debe editarse el comando del D.O.S. deseado. Este debe ser el último de la línea, ya que si bien al editarla no daría error, por el contrario al ejecutarla aparecería el mensaje de error:

```
C Nonsense in BASIC
```

este error también se visualiza si la sintaxis del comando no es correcta.

APLICACIONES

Veamos a continuación un par de aplicaciones sencillas que aclaran un poco más la utilización de estos comandos desde BASIC:

a) Pasar un programa BASIC de cinta a diskette.

Mediante comandos directos podemos ejecutar:

```
LOAD "":RANDOMIZE
   USR 15363:REM:
SAVE "nombre"
```

b) Borrar los ficheros de un diskette.

Cuando se realiza un programa, normalmente tenemos que generar varios ficheros hasta que éste se encuentra totalmente depurado. Una forma racional a la hora de salvarlos, es que su nombre esté formado por una serie de caracteres comunes y el resto que identifique la versión del mismo, por ejemplo: "SECT.1", "SECT.2",...

Cuando tengamos la versión final, deberemos borrar todas laas intermedias. Un método rápido para no tener que borrar uno a uno, lo cual podría ser terrible, sería, suponiendo que deseamos eliminar las siete primeras versiones del ejemplo anterior:

```
10 FOR n=1 TO 7
20 RANDOMIZE USR
   15363:REM:ERASE
   "SECT."+STR$ n
30 NEXT n
```

CODIGOS DE ERROR

Cuando manejamos ficheros desde el D.O.S. cualquier operación errónea, como por ejemplo: borrar un fichero no existente, salvar un programa con un nombre idéntico a otro,..., se nos indicaba mediante el correspondiente mensaje de error: No File(s), File exists,... Por el contrario, cuando se manejan desde BASIC no obtenemos ninguna información en pantalla, no sabiendo por tanto si el comando se ha ejecutado o no. Sin embargo, en el par de registros "BC" del microprocesador Z80 se almacenan unos códigos, correspondientes cada uno de ellos a un tipo de error. En la tabla I se encuentra la lista completa con el significado de estos códigos.

Pero, ¿cómo acceder desde el BASIC a estos códigos? La respuesta está en la utilización de la función "USR". Aunque normalmente va acompañada del comando "RANDOMIZE", puede utilizarse también con "PRINT", "LET", "BORDER", etc.; en estos casos al retornar al BASIC, el contenido del par de registros "BC" se visualiza, queda asignado a una variable o es utilizado para asignar el color del borde.

Ejemplos:

a) Visualización del código.

```
PRINT USR 15363:
   REM:LOAD "green"
   CODE
```

b) Color del borde según el código de error.

```
BORDER USR 15363:
   REM:ERASE "halt"
```

c) Asignación a una variable numérica.

```
LET error=USR 15363:
   REM:NEW "asdf",
   "qwerty": GOSUB
   4000
```

Este método es el utilizado en el programa número 1 (SECTCOD1) cada vez que se realiza una llamada al D.O.S. En la variable "error" se almacena el correspondiente código, y la subrutina de la línea 4000 se encarga de visualizarlo.

El programa es un sencillo *monitor de disco* que permite visualizar y modificar cualquier programa en código máquina por sectores. Ha sido dividido en dos partes, con esta primera se puede acceder por menú a diversos comandos del D.O.S.; en un próximo artículo se complementará con otro programa que deberá combinar con "MERGE".

Código	Error
0	OK
1	Fichero no existente
2	Fichero ya existente
3	Diskette lleno
4	
5	Rebasado núm. sector
6	
7	Diskette defectuoso
8	Sintaxis incorrecta

PROGRAMA 1

```
  10 REM [illegible]
  20 BORDER 1: PAPER 1: INK 7: C
LS
  30 POKE 23658,8
  40 CLS : PRINT "[illegible]"
  50 PRINT AT 6,6;"■ COMANDOS D
.O.S.";AT 8,6;"■ VISUALIZAR SEC
TOR";AT 10,6;"■ SALVAR SECTOR";
AT 12,6;"■ RETORNO AL BASIC";AT
 14,6;"■ RETORNO AL D.O.S."
  60 PRINT #0;AT 1,6;"SELECCIONA
 UNA OPCION"
  70 IF INKEY$="C" THEN GO TO 10
00
  80 IF INKEY$="V" THEN GO TO 20
00
  90 IF INKEY$="S" THEN GO TO 30
00
 100 IF INKEY$="B" THEN CLS : ST
OP
 110 IF INKEY$="D" THEN CLS : RA
NDOMIZE USR 15360
 120 GO TO 70
1000 REM [illegible]
1005 POKE 23658,8
1010 CLS : PRINT "[illegible]"
1020 PRINT AT 6,6;"■ DIRECTORIO
";AT 8,6;"■ BORRAR FICHEROS";AT
 10,6;"■ RENOMBRAR FICHEROS";AT
 12,6;"■ ORGANIZAR FICHEROS";AT
 14,6;"■ MENU PRINCIPAL"
1030 PRINT #0;AT 1,6;"SELECCIONA
 UNA OPCION"
1040 IF INKEY$="D" THEN CLS : GO
 TO 1100
1050 IF INKEY$="B" THEN CLS : GO
 TO 1200
1060 IF INKEY$="R" THEN CLS : GO
 TO 1300
1070 IF INKEY$="O" THEN CLS : GO
 TO 1400
1080 IF INKEY$="M" THEN CLS : GO
 TO 30
1090 GO TO 1040
1100 REM [illegible]
1102 LET error=USR 15363: REM :
CAT
1110 GO SUB 4000: GO TO 1000
1200 REM [illegible]
1202 PRINT AT 1,0;"NOMBRE DEL FI
CHERO: ";
1210 INPUT ">>> "; LINE A$
1212 IF A$="" OR LEN A$>8 THEN G
O TO 1210
1220 PRINT INVERSE 1;A$; INVERSE
 0;"[illegible]"
1230 LET error=USR 15363: REM :
ERASE A$CODE
1240 GO SUB 4000: GO TO 1000
1300 REM [illegible]
1302 PRINT AT 1,0;"NOMBRE ACTUAL
"
1310 INPUT ">>> "; LINE A$
1320 IF A$="" OR LEN A$>8 THEN G
O TO 1310
1330 PRINT INVERSE 1;A$; INVERSE
 0;"[illegible]"
1340 PRINT AT 3,0;"NUEVO NOMBRE
"
1350 INPUT ">>> "; LINE B$
1360 IF B$="" OR LEN B$>8 THEN G
O TO 1350
1370 PRINT INVERSE 1;B$; INVERSE
 0;"[illegible]"
1380 LET error=USR 15363: REM :
NEW B$,A$CODE
1390 GO SUB 4000: GO TO 1000
1400 REM [illegible]
1402 PRINT AT 1,0;" ESPERA UN PO
CO"
1410 LET error=USR 15363: REM :
MOVE
1420 GO SUB 4000: GO TO 1000
3999 STOP
4000 REM [illegible]
4010 IF error<0 OR error>8 THEN
PRINT #0;AT 1,5;"!!! FALLA EL D.
O.S. !!!": GO TO 4050
4020 IF error=0 THEN PRINT #0;AT
 1,0;"[illegible] PULSA ""ENTER""": GO
 TO 4050
4030 RESTORE 5000+error: READ Z$
4040 PRINT #0;AT 1,0; INVERSE 1;
"ERROR ";error; INVERSE 0;" ";Z$
4050 IF CODE INKEY$<>13 THEN GO
TO 4050
4060 RETURN
5000 REM [illegible]
5001 DATA "FICHERO NO EXISTENTE"
5002 DATA "FICHERO YA EXISTENTE"
5003 DATA "DISKETTE LLENO"
5004 DATA "????????????????????"
5005 DATA "REBASADO NUM. SECTOR"
5006 DATA "????????????????????"
5007 DATA "DISKETTE DEFECTUOSO"
5008 DATA "SINTAXIS INCORRECTA"
```

CONSULTORIO

Código máquina

Desearía que me solventéis una duda relativa al código máquina: Hice un pequeño programa de tipo ejercicio para simular el nemónico «ADC BC,A» que, como sabréis, no existe en el assembler del Z-80. El programa es como sigue:

```
LD B,112
LD C,2
LD A,255
ADD A,C
LD C,A
LD A,Ø
ADC A,B
LD B,A
RET
```

Hasta este punto, todo bien, pero el problema se me planteó al querer hacer un programa similar que imitara la resta: «SBC BC,A» teniendo «A» y «BC» los mismos valores que en el ejemplo anterior. No tengo ni idea de cómo realizarlo.

Francisco J. LOPEZ - Barcelona

☐ En el programa que nos manda, da la impresión de que se complica usted demasiado; suponemos que desea preservar el valor de los restantes registros; no obstante, la cosa se hace más sencilla usando «HL» y para preservarlo tenemos la pila:

```
PUSH HL
PUSH BC
POP HL
LD B,Ø
LD C,A
ADC HL,BC
PUSH HL
POP BC
POP HL
```

Es una forma un tanto «rara» de programar, pero hace exactamente lo que usted pide (aunque no sabemos para qué puede usarlo). Si desea restar, simplemente cambie la instrucción «ADC HL,BC» por «SBC HL,BC» y problema resuelto. En general, le será más cómodo utilizar como acumulador, en operaciones aritméticas de 16 bits, el registro «HL» en lugar del «BC», ya que el microprocesador tiene pensado el juego de instrucciones para que se haga de esta forma.

Interioridades del Sistema Operativo

¿Por qué la dirección 23613 es una de las direcciones «sagradas» del Spectrum? Ya sé que es la dirección del elemento de la pila de máquina que es usado como retorno de error, pero no sé qué significa ni para qué sirve.

¿Para qué sirven las direcciones 23755 y 23568?

Rafael MESA - Valencia

☐ Es casi imposible responder a sus preguntas sin adentrarse excesivamente en el funcionamiento del Sistema Operativo, lo cual, daría para escribir un libro entero (quizá algún día lo escribamos). No obstante, vamos a intentar explicárselo aún a riesgo de que no se entere de nada.

Cuando el ordenador está ejecutando un programa en Basic, es posible que se produzca un error, en ese caso, hay que interrumpir la ejecución y retornar al editor presentando, en pantalla, un mensaje de error. En el punto donde se detecta el error, es posible que se hayan hecho ya varias llamadas a subrutinas (CALL), con lo que la pila estará llena de direcciones de retorno; si quisieramos volver al editor, tendríamos que hacer tantos RET como CALL hubiéramos hecho, el problema es que no hay forma de saber cuántos son. El sistema utiliza un curioso procedimiento: después del primer CALL, se guarda en ERR_SP el contenido del registro «SP», de forma que, cuando queramos retornar al editor, no tenemos más que cargar en el registro «SP» el contenido de ERR_SP y nos habremos saltado toda la pila de una vez, encontrándonos en la misma situación que después del primer CALL; ahora no hay más que hacer un RET y estaremos en el editor.

Las otras dos direcciones, tienen que ver con los canales de comunicación, 23755 cae dentro de la tabla de canales y 23568 es la dirección base de la tabla de corrientes. Vamos a intentar ver qué son los canales y las corrientes.

En su configuración básica, el Spectrum puede comunicarse por cuatro canales, estos son: «R», «S», «K» y «P»; «R» es el área de trabajo, «S» es la parte superior de la pantalla, «K» es el teclado y la parte inferior de la pantalla, y «P» es la impresora. En los comandos del Basic, podemos definir hasta 15 corrientes de comunicación, se denominan desde #1 hasta #15; el INPUT es, siempre, INPUT #1, el PRINT es PRINT #2 y el LPRINT es PRINT #3; por tanto, la corriente #1 está unida al canal «K», la #2 lo está al canal «S» y la #3 al canal «P». Podemos hacer OPEN #2; «P» y todas las sentencias PRINT se dirigirán a la impresora, habremos unido el canal «P» con la corriente #2. Pero, ¿cómo sabe el ordenador a qué canal está unida cada corriente? La solución es que se utilizan dos tablas; la primera se denomina STRMS y su dirección base es 23568; en esta tabla, se utiliza el número de corriente como un «offset» y se sale con otro «offset», que nos sirve para entrar en la tabla de canales; la dirección base de esta segunda tabla está contenida en la variable CHANS y, si le sumamos el «offset» que hemos leído en STRMS, estaremos apuntando a las direcciones de entrada y salida que corresponden al canal seleccionado.

Evidentemente, es imposible entender todo lo anterior si no se sabe Assembler; si es su caso, le remitimos a nuestro Curso de Código Máquina donde explicamos con todo detalle el manejo de la pila y la forma de utilizar tablas de «offset».

«PAUSE Ø»

Necesito saber si existe alguna rutina de la ROM (o alguna otra que pueda reubicarse) que lea el teclado y detenga el programa que esté pulsada alguna tecla, con lo que continuaría el programa sin devolver el control al Basic.

Antonio TORREGROSA - Cádiz

☐ Hay varias formas de hacer lo que usted pide, una de ellas sería: hacer un «HALT» para esperar por una interrupción y, luego, comprobar el bit 5 de «FLAGS» que será «1» si se ha pulsado alguna tecla y «Ø» en caso contrario; podría ser algo así:

```
KEY     HALT
        BIT  5,(FLAGS)
        JR   Z, KEY
        RES  5,(FLAGS)
        RET
FLAGS EQU 23611
```

Si tiene cambiado el vector de interrupción, utilice «RST #38» en lugar de «HALT».

Puesto que lo que usted quiere hacer es algo similar a un «PAUSE Ø», ¿por qué no utilizar la rutina «PAUSE» de la ROM con parámetro «Ø»?:

```
KEY        LD   BC,Ø
           CALL PAUSE-1
           RET
PAUSE-1 EQU 7997
```

«DISCO-ROM»

Respecto a los artículos sobre el «DISCO-ROM», les

quiero preguntar lo siguiente:

¿El circuito impreso y los componentes van a ser distribuidos por ustedes de alguna manera o cada uno se los tiene que procurar como pueda?

¿Pondrán a la venta EPROMs ya grabadas o sólo los programas y la grabación la deberá hacer el usuario? Lo primero sería muy interesante y una experiencia totalmente nueva en informática personal.

Alberto A. CIFRIAN - Cantabria

□ Nuestro trabajo consiste en la creación de revistas, no de «Kits»; en algún caso —como el del Joystick— nos hemos lanzado a la distribución de un kit completo, pero esto sólo es económicamente posible si se trata de un montaje de amplia difusión (miles de compradores), en caso contrario, la tarea está abocada al fracaso.

Respecto a la idea de comercializar EPROMs grabadas, es un tema en el que ya habíamos pensado, no en vano, HOBBY-PRESS se presta de ser una editorial innovadora en su campo (recuerde el «DISCO-FLEX» y la cinta «HOBY-SUERTE» que supusieron en su día un inmenso esfuerzo editorial); no obstante, aparece el mismo problema de antes, las EPROM son caras y, cada una, tarda casi un minuto en grabarse, por lo que su distribución sólo es posible si se espera un número alto de compradores. De momento, esperaremos a ver la aceptación que tiene el «DISCO-ROM» entre nuestros lectores.

Copia de programas

¿Cómo se pueden sacar los códigos de un programa para poderlo grabar?

Me gustaría saber si se podría grabar un programa de cassette a cassette con las clavijas del Spectrum o con algún cable especial.

Oscar FERNANDEZ - Cantabria

□ Suponemos que lo que usted quiere hacer es sacar una copia de seguridad de sus programas; para ello, necesitará desprotegerlo, preceso que no es nada sencillo si no se tiene un gran conocimiento del Spectrum.

La segunda solución que propone, se denomina «copia analógica» y no siempre es posible realizarla; depende, fundamentalmente, del tipo de cassettes empleados.

En todo caso, le recordamos que la obtención de copias de seguridad es una actividad lícita sólo si son para uso propio; la distribución de estas copias constituiría un delito contra la propiedad intelectual, y ya hay quien ha acabado en la cárcel por ello.

La avería más típica

Poseo un ordenador Spectrum Plus de referencia ISSUE 6A comprado en Inglaterra, y presenta la siguiente avería:

Al proporcionar alimentación al aparato, este visualiza por el televisor el típico rectángulo de 23 x 31 caracteres formando algunas columnas blancas y negras, alternadas, salpicando la pantalla de caracteres de color.

Abierto el ordenador, sólo he podido constatar la presencia de las tensiones aparentemente correctas y de un sospechoso calentamiento en la ULA que es del tipo Ferranti 6COO1E-7/8447.

Desearía saber si esta avería puede provenir exclusivamente de la ULA o si el calentamiento se debe a un funcionamiento indebido de algún otro componente.

Asimismo, sospecho que la avería esté motivada por la conexión de un interface para joystick Kempston con el ordenador en funcionamiento.

Joaquín R. FIGUERAS - Barcelona

□ Efectivamente, si ha conectado un interface con el ordenador funcionando, no le de más vueltas, la avería viene de ahí. Se trata de la avería más frecuente del Spectrum, por esa razón, hemos insistido tanto en que nunca se conecte ningún interface mientras el ordenador esté funcionando.

El calentamiento de la ULA es normal; lo más probable es que funcione bien ya que, si no, no aparecería imagen alguna en la pantalla. Lo más indicado es que empiece por cambiar el Z-80, es el componente más barato (menos de 2.000 ptas.) y el que, con mayor probabilidad, está averiado; utilice un desoldador para desoldar las 40 patillas, y no lo aplique durante mucho tiempo a cada una o correrá el riesgo de destruir la placa de CI. Antes de montar el nuevo Z-80, coloque un zócalo que le facilitará posteriores reparaciones. Si este cambio no resolviera el problema, la cosa podría venir de la ULA, en cuyo caso, sería mejor que enviara el ordenador a un taller, ya que las ULAs son componentes difíciles de conseguir.

DE OCASION

● VENDO Spectrum Plus, con garantía, alimentador, estabilizador, cassette, revistas, ampli-sonido, etc. Todo por 37.000 ptas. Antonio. Tel. (974)603179. Rda. Turia, 14, 4.° Teruel.

● DESEO entrar en contacto con algún club de usuarios del Spectrum, con residencia en Madrid. Mi dirección es la siguiente: José Pajuelo Ramos. Tituleia, 21. Getafe (Madrid).

● VENDO Spectrum 48k, totalmente nuevo, con fuente de alimentación, cables, cassette de demostración y dos manuales por sólo 27.000 ptas. Interesados llamar al Tel. 4235232. Madrid.

● VENDO Zx Spectrum 48k, con fuente de alimentación, cables, cassette de demostración y garantía. Sólo por 28.000 ptas. Interesados escribir a Fco. Javier Muñoz Romero. Prolongación de Colón, 3. Santa Fe (Granada).

● INTERESADOS en formar parte de un club de usuarios de ZX, para intercambiar ideas, listados, etc. Escribir a José Angel Soriano Rodríguez. San Emigdio, 75. Almoradí (Alicante), o bien llamar al Tel. (965)700508.

● VENDO Spectrum 16k, por sólo 19.900 ptas. Interesados llamar al Tel. (94)4580962 ó bien dirigirse a Oscar Casares. Erdiko, 9, 4.° dcha. Alonsótegui (Vizcaya).

● VENDO Interface AGF Protocol (es a la vez Kempston y programable), por sólo 2.000 ptas. Interesados llamar al Tel. (94)670255 de Vizcaya o bien escribir a Iñigo. Mendietas Zeharbide, 6, 2.° a. Sopilana (Vizcaya).

● DESEARIA entrar en contacto con usuarios del Spectrum 16/48k de la zona de San Sebastián y alrededores para el intercambio de ideas, etc. Interesados llamar al Tel. (943)514090 y preguntar por Miguel o bien al Tel. 520132. Pablo. Dirección: San Marcos, 5, 3.° A. Rentería (Guipúzcoa).

● VENDO Spectrum 48k. Magnífico estado, fuente de alimentación, manuales, cinta Horizontes. Interface para 2 joystick, Interface II o Kempston con salida monitor, libros, revistas, etc. Todo por 35.000 ptas. Preguntar por Javier al Tel. 8582231 ó bien escribir a Javier Sanz Terrada. Apartado de Correos, 55. Las Rozas (Madrid).

● VENDO TRC casi nuevo, precio en la calle sobre las 13.000 ptas. compuesto por 8 rectas, 8 curvas, 1 transformador 125-220 A, 17,2 VI, 2 mandos, 1 trampolín obstáculo, 2 coches y además regalo 1 coche, todo ello por el precio de 5.000 ptas. o lo cambio por interface programable más Quick Shot 2. Interesados llamar al Tel. (91)4292823. Preguntar por Victor.

● VENDO Spectrum 48k y sus accesorios. Precio 25.000 ó 28.000 ptas. Interesados llamar al Tel. (93)3594909 ó bien escribir a Pedro Hinajeros. Casas y Amigó, 66, A-2, Entlo 2.ª. 08016 Barcelona.

● DESEARIA contactar con usuarios del Spectrum 16/48k, para intercambiar ideas, trucos, información y formar un grupo par aprogramar. Sólo en Lugo. Interesados dirigirse a José Arias. Ruiseñor, 19, 1.° izda. Lugo. Tel. 212045.

● VENDO Joystick Quick Shot 1 e Interface kemspton. Junto por 2.500 ptas. y por separado 1.500 ptas. Vendo tambié al precio de 850 ptas. unidad, instrucciones. Interesados contactar con Pedro Vera de la Rocha. Rbla. San Estaban, 11. Ripollet (Barcelona). Tel. 6922253.

● VENDO Spectrum 48k, con todos sus accesorios, 2 libros y bastantes revistas. Precio 30.000 ptas. P. Miguel. Tel. 3095874 de Barcelona.

● URGE vender Spectrum Plus, con seis meses de garantía, a estrenar, por sólo 35.000 ptas. con todos sus cables, transformador con interruptor, libros, etc. También desearía comprar Impresora Sp-800 de Seikos. Interesdos escribir Carlos Bermejo Aguilar. P° Zorrilla, 67. Valladolid.

● VENDO Spectrum 48K, en perfecto estado, con teclado profesional DK'Tronics, interruptor ON-OFF e interruptor Reset instalados, lápiz óptico a estrenar, manuales en castellano. Precio 40.000 ptas. Regalo revistas y teclado original del Spectrum. Interesados llamar al Tel. 4563064 o bien escribir a Antonio Hernández Sánchez. Capitán Haya, 39. Madrid.

● VENDO Interface joystick tipo Kempston y Quick Shot II, por 4.500 ptas. Preferentemente de León. Llamar al Tel. (987)204380 (a partir de las 22 h). Preguntar por César.

● VENDO libros: «Manual para principiantes de informática», «Cómo crear tus propios juegos», por 500 ptas. cada uno. Llamar al Tel. (981)316869. Carlos.

● VENDO video-juegos marca Supersportic. 6 ó 7 cartuchos, por 7.000 ptas. o bien lo cambio por un micrófono con eco. Apartado 888. Granada.

● VENDO Spectrum 48K, en perfecto estado por dentro y exterior, con numerosas revistas. Precio a convenir. Carlos López. Ceñorio de Vizcaya, 19, 4.°. Guernica. Tel. 68515115 (a partir de las 8 h).

● VENDO ordenador Spectravideo 328, monitor fósforo verde, expander 601 portatarjetas, cassette 904, tarjeta centronic, cable impresora por 7.500 ptas. J. Marimón. Tel. 2410325 por las tardes. Barcelona.

● VENDO ordenador Láser 310, teclado profesional, completo, con libros de instrucciones. Precio 25.000 ptas. Interesados llamar al Tel. (93)3527641.

● CLUB de software ampliamos socios. Vendemos ampliación externa en buen estado por 6.000 ptas. Interesados escribir a Joaquín Compañ. Ramón y Cajal, 24. Villajoyosa. Alicante.

● VENDO por cambio de ordenador una ampliación externa a 48K, por sólo 8.000 ptas. Información al Tel. 2397917 de Barcelona.

● DESEO tener correspondencia acerca del Spectrum. Interesados escribir a Carles Jordi Fernández. Ctra. a Bagá, 42. Guardiola de Bergueda. Barcelona.

● VENDO Spectrum 16K y revistas Microhobby del 1 al 25. Precio 20.000 ptas. Tel. (96)7221658 (9 a 3 h). Preguntar por Charo.

● CLUB Comespectrum 112K. Usuarios del CBM 64 y Spectrum 48K. Pide información llamando al Tel. (972)330669 ó 333819. Preguntar por Paco. Gerona.

● VENDO Spectrum 48K, con cables, manuales, libros C/M y cóbol por tan sólo 22.000 ptas. (Negociables). Tiene sólo seis meses. También vendo, por su precio de adquisición, las colecciones completas de Microhobby, ZX y Todospectrum. Llamar al Tel. (93)3718518. Preguntar por Alfonso (a partir de las 22 h).

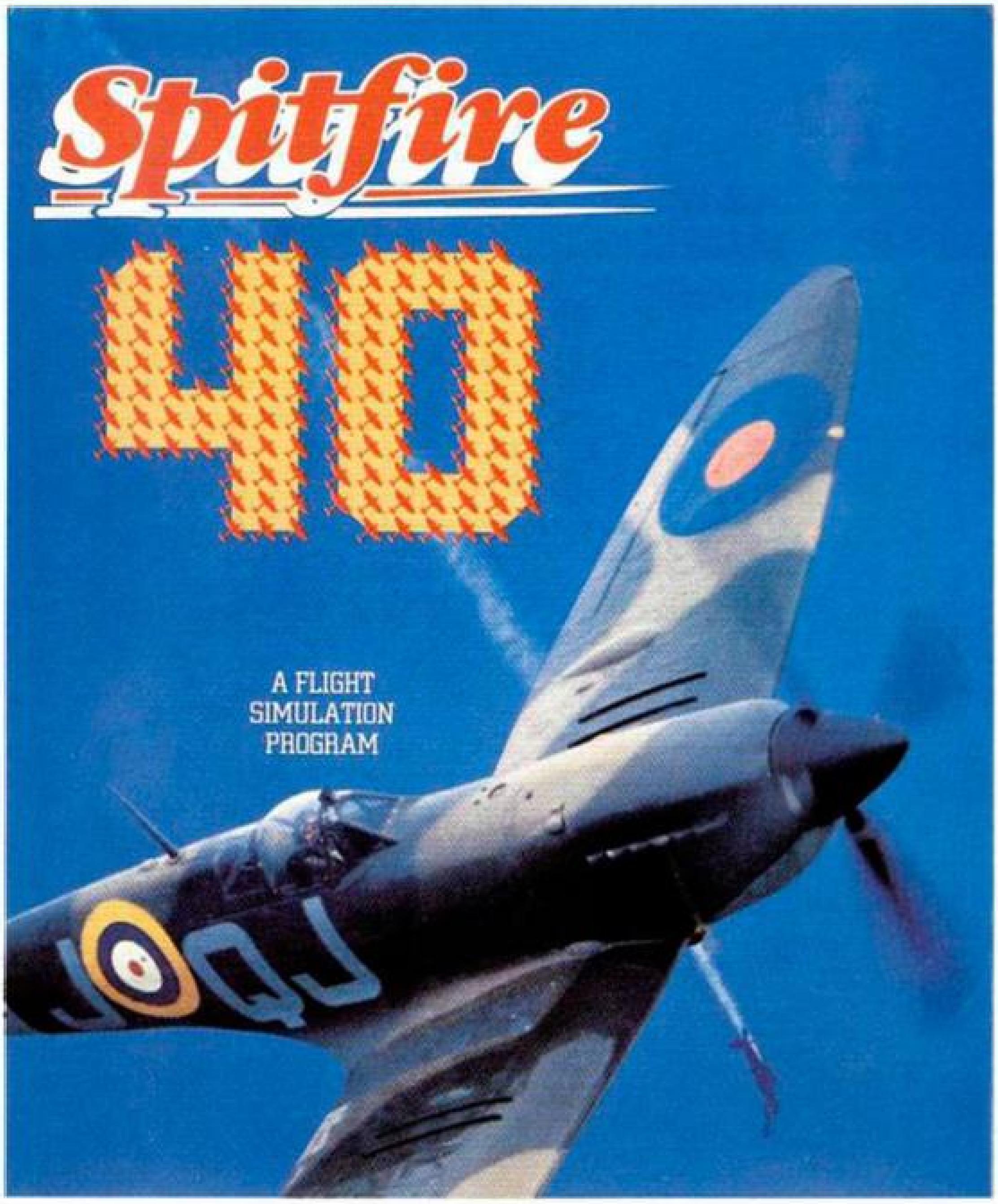
● VENDO ampliación externa de memoria, 32K para Spectrum, por 8.000 ptas. Interesados escribir a Antonio Canillas Galiano. Arango, 3, 6.° A. Málaga o bien llamar al Tel. (952)304217. Preguntar por Antonio.

● DESEARIA comprar un microprocesador Júpiter Ace que esté en buenas condiciones de uso (incluyendo manual, transformador de corriente y cables) que sea económico. Si además tiene algún periférico mejor. También compraría un monitor de 9 a 12 pulgadas (muy económico). Ofertas dirigirlas a Fco. José Alvarez González. Juan Alvargonzález, 21, 4.° Izqda. Gijón (Asturias). Tel. (985)394598. Llamar a partir de las 9,30 h.

● COMPRO fuente de alimentación original para ZX Spectrum. Interesados, preferiblemente de Barcelona, llamar al número (93)2571685 (horas de comida) y preguntar por Andrés.

ZAFI CHIP

te ofrece el mejor programa de simulación de vuelo creado hasta la fecha.



SPECTRUM 48k
COMMODORE 64

El Spitfire 40 no es sólo el mejor simulador de vuelo de los aviones más famosos de todos los tiempos, también es una espectacular aventura de guerra.
La escena es el verano de 1940 y tú eres un piloto recién entrenado, destinado en un Escuadrón Spitfire en algún lugar del Sureste de Inglaterra. Como uno de tantos jóvenes en 1940, aprenderás que un Spitfire no es un avión corriente, descubrirás sus especiales capacidades y lo más importante de todo, cómo manejarlo durante el combate.

Si deseas información y participar en los importantes sorteos que ZAFICHIP celebrará durante el año. . . ¡ESCRIBENOS!

Si están agotados en tu tienda habitual ¡¡LLAMANOS!!

ZAFIRO SOFTWARE DIVISION
Paseo de la Castellana, 141. 28046 Madrid
Tel. 459 30 04. Tel. Barna. 209 33 65. Télex: 22690 ZAFIR E


SI BUSCAS LO MEJOR ERBE Software LO TIENE
CYBERUN
LO ULTIMO DE
ULTIMATE PLAY THE GAME
¡UN RETO A TU FANTASIA¡
DISTIBUIDOR EXCLUSIVO PARA ESPAÑA ERBE SOFTWARE C/. STA. ENGRACIA, 17. 28010 MADRID, TFNO.: (91) 447 34 10
DELEGACION BARCELONA, AVDA. MISTRAL, N.º 10 - TFNO.: (93) 432 07 31